



Universidad  
Carlos III de Madrid

Grado en Ingeniería en Tecnologías de Telecomunicación  
Curso académico (2015-2016)

*Trabajo Fin de Grado*

***Diseño e implementación de mecanismos de  
autenticación basados en Eyetracking***

Autor

Jorge Álvaro Garrido Cortés

Tutora

Patricia Arias Cabarcos

Presentación: 4 de octubre de 2016.



## Resumen

Durante las últimas décadas las tecnologías han experimentado un gran avance, provocando que cada vez sea mayor el número de servicios a los que los usuarios pueden acceder de forma electrónica. Debido a esto, surge la necesidad de buscar soluciones y mejoras en la interacción persona-máquina.

Paralelamente, con el aumento de los delitos de intrusión informática que hay actualmente, se busca mejorar la seguridad con el fin de disminuir las probabilidades de que un delincuente pueda acceder a un dispositivo ajeno. En especial, se desea mejorar la primera y más importante barrera de seguridad, la autenticación.

Al ser la usabilidad un aspecto crucial para la adopción y buen uso de las tecnologías de seguridad, en este trabajo, se ha decidido estudiar e implementar un algoritmo de autenticación con una nueva tecnología llamada *eyetracking*. Dicha tecnología se basa en la capacidad de calcular el lugar exacto que una persona está mirando a través de información extraída de su cara y ojos, lo que aportará una gran facilidad de uso y seguridad.

Haciendo uso del dispositivo y el API que ofrece The Eye Tribe, se ha desarrollado un *software* que, empleando los datos proporcionados por dicho dispositivo, permite autenticar a un usuario a través del movimiento de su ojo frente a unas imágenes.

**Palabras clave:** eyetracking, autenticación, seguridad, usabilidad, software



## Índice general

|   |           |
|---|-----------|
| <b>Resumen .....</b>  | <b>3</b>  |
| <b>Índice general.....</b>  | <b>5</b>  |
| <b>Índice de tablas.....</b>  | <b>7</b>  |
| <b>Índice de ilustraciones.....</b>                                 | <b>9</b>  |
| <b>I. Introduction.....</b>   | <b>11</b> |
| I.1 Motivation and goals.....                                       | 12        |
| I.2 Development stages.....   | 14        |
| I.3 Employed means.....   | 15        |
| I.4 Memory structure.....   | 19        |
| <b>II. Extended abstract .....</b>                                  | <b>21</b> |
| <b>III. Conclusions.....</b>  | <b>27</b> |
| III.1 Conclusions.....  | 27        |
| III.2 Future work lines.....  | 27        |
| <b>Capítulo 1: Introducción.....</b>                                | <b>29</b> |
| 1.1 Motivación y objetivos.....                                     | 30        |
| 1.2 Fases de desarrollo .....                                       | 32        |
| 1.3 Medios empleados .....  | 33        |
| 1.3.1 Eyetracker de The Eye Tribe.....                              | 34        |
| 1.3.2 The Eye Tribe SDK 0.9.56 .....                                | 35        |
| 1.3.4 Entorno de desarrollo NETBEANS.....                           | 37        |
| 1.3.4 Librerías: The Eye Tribe y Otras .....                        | 38        |
| 1.4. Estructura de la memoria .....                                 | 38        |
| <b>Capítulo 2: Planteamiento del problema.....</b>                  | <b>41</b> |
| 2.1 Análisis del estado del arte .....                              | 41        |
| 2.1.1 Tecnologías de seguimiento de ojos o <i>Eyetracking</i> ..... | 41        |
| 2.1.2 Técnicas de autenticación basadas en <i>Eyetracking</i> ..... | 51        |
| 2.2 Restricciones y marco regulador.....                            | 53        |
| <b>Capítulo 3: Análisis y diseño de la solución técnica.....</b>    | <b>55</b> |
| 3.1 Especificación de requisitos .....                              | 55        |



|  |           |
|--|-----------|
| 3.2 Arquitectura.....                                  | 56        |
| <b>Capítulo 4: Implementación .....</b>                | <b>61</b> |
| <b>Capítulo 5: Evaluación y resultados.....</b>        | <b>71</b> |
| 5.1 Dificultades en el desarrollo .....                | 71        |
| 5.2 Pruebas realizadas.....                            | 71        |
| <b>Capítulo 6: Gestión del Proyecto.....</b>           | <b>75</b> |
| 6.1. Planificación.....                                | 75        |
| 6.2 Presupuesto.....                                   | 76        |
| <b>Capítulo 7: Conclusiones y líneas futuras .....</b> | <b>79</b> |
| 7.1 Conclusiones.....                                  | 79        |
| 7.2 Líneas futuras.....                                | 79        |
| <b>Referencias.....</b>                                | <b>81</b> |

## Índice de tablas

|   |           |
|---|-----------|
| <b>Table 1: Comparative of technical features of every eye-tracking Device .....</b>      | <b>22</b> |
| <b>Tabla 2: Características técnicas del tracker The Eye Tribe .....</b>                  | <b>43</b> |
| <b>Tabla 3: Características técnicas del tracker SentiGaze .....</b>                      | <b>45</b> |
| <b>Tabla 4: Características técnicas del PC .....</b>                                     | <b>45</b> |
| <b>Tabla 5: Características técnicas WebCam .....</b>                                     | <b>46</b> |
| <b>Tabla 6: Características técnicas del tracker AEye Eye .....</b>                       | <b>46</b> |
| <b>Tabla 7: Características técnicas del tracker Tobii .....</b>                          | <b>47</b> |
| <b>Tabla 8: Características técnicas del tracker de EyeGaze .....</b>                     | <b>49</b> |
| <b>Tabla 9: Comparativa genérica de diferentes productos de Eye-Tracking .....</b>        | <b>50</b> |
| <b>Tabla 10: Comparativa prestaciones técnicas de cada producto de Eye-tracking .....</b> | <b>50</b> |
| <b>Tabla 11: Asociación del fichero de visualización con el controlador .....</b>         | <b>70</b> |
| <b>Tabla 12: Intervalo de tiempo transcurrido por tarea .....</b>                         | <b>76</b> |
| <b>Tabla 13: Cantidad de horas dedicadas por tarea .....</b>                              | <b>77</b> |
| <b>Tabla 14: Desglose de los costes materiales del Proyecto .....</b>                     | <b>77</b> |
| <b>Tabla 15: Desglose de los costes personales del Proyecto .....</b>                     | <b>78</b> |
| <b>Tabla 16: Desglose de los costes totales del Proyecto .....</b>                        | <b>78</b> |





## Índice de ilustraciones

|  |    |
|--|----|
| Illustration 1: Person-machine interaction with Eyetracking technology .....   | 11 |
| Illustration 2: Example of authentication by drawing a pattern .....           | 13 |
| Illustration 3: Eye tracker of The Eye Tribe .....                             | 16 |
| Illustration 4: Example of a heat map in a website .....                       | 16 |
| Illustration 5: Example of usability in a tablet .....                         | 17 |
| Illustration 6: Java's Logotype .....  | 18 |
| Illustration 7: Logotipo de NetBeans .....                                     | 18 |
| Illustration 8: Diagram of the different authentication techniques .....       | 22 |
| Illustration 9: Application architecture .....                                 | 23 |
| Illustration 10: Calibration window.....                                       | 24 |
| Illustration 11: Main window .....   | 24 |
| Illustration 12: Evaluation window.....  | 24 |
| Illustration 13: Password window.....  | 25 |
| Illustration 14: Log-in window.....  | 25 |
| Ilustración 15: Interacción persona-máquina con la tecnología .....            | 29 |
| Ilustración 16: Ejemplo de autenticación mediante el dibujo de un patrón ..... | 31 |
| Ilustración 17: Eye tracker de The Eye Tribe .....                             | 34 |
| Ilustración 18: Ejemplo de un heat map en una página web .....                 | 35 |
| Ilustración 19: Ejemplo de usabilidad en una tableta .....                     | 35 |
| Ilustración 20: Logotipo de Java .....   | 36 |
| Ilustración 21: Logotipo de NetBeans .....                                     | 37 |
| Ilustración 22: Logotipo The Eye Tribe .....                                   | 43 |
| Ilustración 23: Tracker de The Eye Tribe .....                                 | 44 |
| Ilustración 24: Logotipo Neurotechnology .....                                 | 45 |
| Ilustración 25: Logotipo EyeTech .....   | 46 |
| Ilustración 26: Tracker de EyeTech .....                                       | 47 |
| Ilustración 27: Logotipo Tobii Experience .....                                | 47 |
| Ilustración 28: Tracker de Tobii .....   | 48 |
| Ilustración 29: Logotipo LC Technologies .....                                 | 49 |

|  |           |
|--|-----------|
| <b>Ilustración 30: Tracker de EyeGaze en un monitor .....</b>                    | <b>49</b> |
| <b>Ilustración 31: Diagrama de las distintas técnicas de autenticación .....</b> | <b>53</b> |
| <b>Ilustración 32: Arquitectura de la aplicación .....</b>                       | <b>56</b> |
| <b>Ilustración 33: Ventana principal .....</b>                                   | <b>57</b> |
| <b>Ilustración 34: Ventana calibration .....</b>                                 | <b>57</b> |
| <b>Ilustración 35: Ventana evaluation .....</b>                                  | <b>58</b> |
| <b>Ilustración 36: Ventana de log-in .....</b>                                   | <b>59</b> |
| <b>Ilustración 37: Ventana password .....</b>                                    | <b>59</b> |
| <b>Ilustración 38: Ventanas de autenticación correcta e incorrecta .....</b>     | <b>60</b> |
| <b>Ilustración 39: Estructura del proyecto en Eclipse .....</b>                  | <b>62</b> |
| <b>Ilustración 40: Estructura de la clase <i>Main</i> .....</b>                  | <b>63</b> |
| <b>Ilustración 41: Estructura del paquete <i>scenes</i> .....</b>                | <b>64</b> |
| <b>Ilustración 42: Estructura del paquete <i>ui</i> .....</b>                    | <b>67</b> |
| <b>Ilustración 43: Estructura del paquete <i>utils</i> .....</b>                 | <b>69</b> |
| <b>Ilustración 44: Área y distancia de funcionamiento del tracker .....</b>      | <b>72</b> |
| <b>Ilustración 44: Diagrama Gantt del TFG .....</b>                              | <b>75</b> |

## I. Introduction

During the last decades, technologies have experienced impressive progress. This is due, mainly, to the strong expansion of internet, causing that increasing the number of services which users can access electronically.

Currently, you can listen to music, watch TV, perform administrative tasks, communicate with others, buy all kind of necessities like food and clothes, play and perform many other functions only through Internet. Because of this extensive list of available services, the need arises to find solutions and improvements in human-machine interaction.

A new technology that can do it is the based on eye tracking, through which it is achieved that the user experience and accessibility are unique, allowing to use the view as if it were a computer mouse, something that would specially benefit people with certain physical or mental disabilities who, finally, would be able to perform data communications tasks for themselves. This technique is also able to improve the architecture of web pages through the study of “heat maps”, make digital marketing studies and even develop applications that provide security to electronic devices.

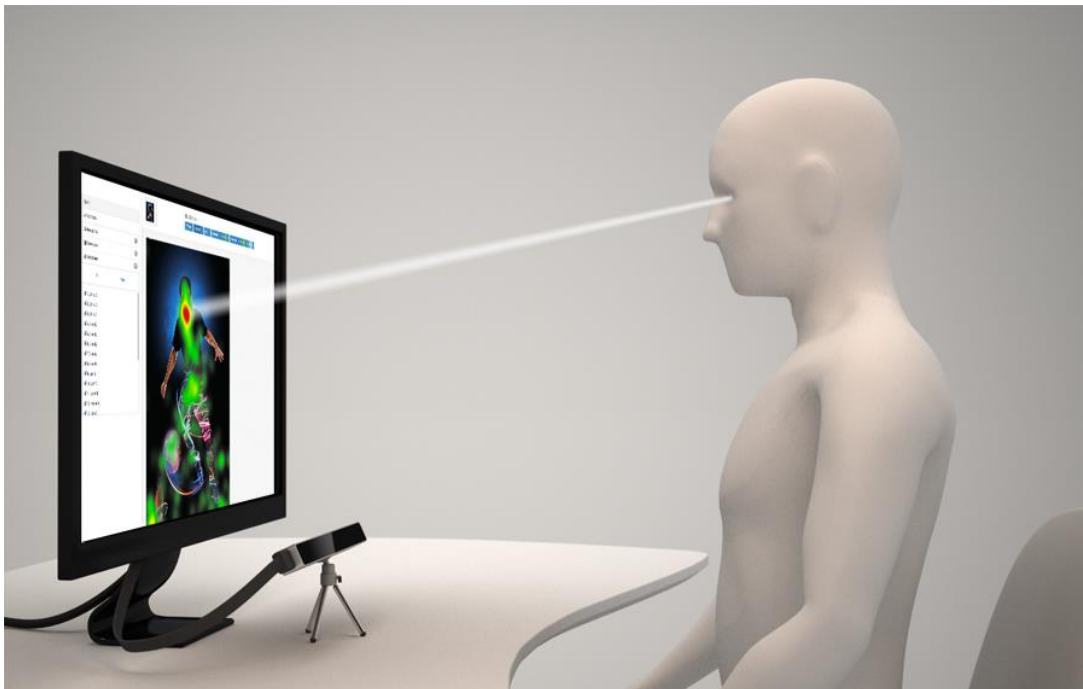


ILLUSTRATION 1: PERSON-MACHINE INTERACTION WITH EYETRACKING TECHNOLOGY

## I.1 Motivation and goals

### MOTIVATION

Today, people save all their files and sensitive data on their computers, smartphones and tablets. Because of this, it requires a strong security to prevent third parties penetrating devices and steal information or infect the terminal.

With the great increase in crimes of this kind in this time, it is seeking to improve safety in order to reduce the probability that a delinquent can access to a device. In particular, it aims to improve the first and most important security barrier, the authentication.

At the same time, it should be noted that, as U. Jendricke said in an article [1] and later proved, usability has always been and always will be a crucial aspect in the adoption and proper use of security technologies.

Currently, several improvements have been proposed to the traditional authentication based on a single factor, that is, one in which an element is known by the user (password or pattern). Among these improvements stand out:

- Those based on **two-factor authentication**, which are a combination of two different authentication methods where there is an element that is known for the user (password or pattern) and one element that the user has (DNI, phone number, etc.)
- The **biometrics-based** consisting on verifying the identity of a person by considering morphological characteristics that are inherent and unique in every person. That is, features such as fingerprint, iris and voice.
- The ones based on a technology called **Eye-tracking**, which is able to calculate the exact place where a person is looking at, throughout the information taken from your face and eyes.

It has been proven that different authentication techniques based on biometrics and eye-tracking technology, provide a greater usability to the user against the ones traditionally used.

Among these techniques that have significant improvements, the one based on eye-tracking stands out due to the fact that it is the most innovative one. This technology allows the implementation of different methods by which a user can be authenticated: drawing patterns with the sight, reading texts, following with his gaze one or more images, etc. These methods (detailed in section 2.1.2) are able to provide a strong security, while they are not very difficult to remember and they are accessible even for people with certain disabilities. On the other hand, the traditional authentication techniques would need some physical movement, driving either a mouse or a finger.

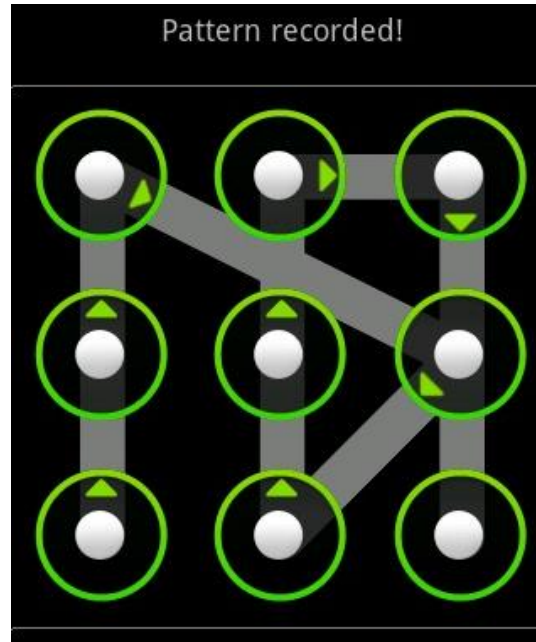


ILLUSTRATION 2: EXAMPLE OF AUTHENTICATION BY DRAWING A PATTERN

Because of the great usability and security that can provide the eye-tracking technology in the authentication techniques, an authentication algorithm will be designed and implemented in order to understand the process of implementation, the benefits presented by the final application and the ease of use and safety that is able to provide.

## GOALS

The overall objective of this paper is to study, implement and evaluate a low cost prototype for authentication based on eye-tracking technology, in order to ensure a better usability and security in the authentication of a user.

An eye tracking device (called tracker) is going to be used. It allows to capture and record the point that a person is looking at when he is in front of a screen. With the data returned by the tracker it is intended to be able to authenticate an individual by a technique that will be chosen during the project.

This Project will try to come up with an authentication method that is innovative and that has not been implemented or tested yet, a method that is also easy to develop and able to provide essential features to the authentication as ease of use and security.

In order to achieve the overall objective and the conclusion of this paper, a number of sub-objectives to follow have to be established. These sub-goals are:

1. **Analyze and compare the different eye-tracking technologies.** A study of different eye-tracking technologies on the market as The Eye Tribe, SentiGaze, Tobii, etc. will be held. After that, they were compared taking into account the technical and functional characteristics of each one and, eventually, the most appropriate solution for this project will be selected.

2. **To study the different authentication techniques.** An analysis of the different authentication techniques that could fit as a solution to the problem will be carried out, and the one that is the most innovative, easy to develop and that is able to provide a great usability and security will be chosen.
3. **To become familiar with the tracker.** Once known its features and its technical characteristics, all of them will be tested in the physical device to examine every offered option and its limitations will be evaluated.
4. **To study the existing libraries.** Once seen the hardware device possibilities, it is necessary to study and understand every function, method and objects available in the API. It will be studied if among these libraries there are the items that are required to carry out the development of the application, the Reading of eyes, their position, the representation of images or points, etc. It will also be essential to analyze the different programming languages available in the SDK so in this way, and depending on which one is the most complete and simple, to choose one of them to implement the algorithm.
5. **To adapt to the programming language and development environment.** Once the programming language has already been decided, the next step is to train in this language and choose a development environment that supports it.
6. **To design and implement an authentication architecture.** Once the authentication technique and the most appropriate programming language have been chosen and the API has been studied, it's time to design the architecture that the application of authentication will have. It will seek for a technique that this is simple to implement, easy to understand by the user and as visual as possible. Subsequently, the implementation of the architecture will be carried out in the previously selected development environment.
7. **To evaluate the performances of the implemented solution compared to traditional mechanisms.** When the algorithm has been developed, various tests to evaluate its benefits and limitations will be performed. Then, it will compare, in terms of performance, the solution obtained against traditional authentication mechanisms. Especially the aspects that have to do with usability and security of the algorithm will be analyzed and finally, a final conclusion will be taken from them.

## I.2 Development stages

For the development of this Final Project a series of five stages are going to be carried out, in order to meet the objectives:

- **Analysis of the technology.** In this stage, the initial project planning is first prepared and then, a wide analysis of the environment, the technologies and the different tools that will be used is performed. To do so, the documentation provided by The Eye Tribe on eye tracking device is studied in depth and the different authentication techniques are compared.

- **Design of the solution.** In this second stage the steps that will be followed in implementation of the solution are defined. In addition, design decisions are taken, such as the authentication technique and the programming language that will be used; and the functional and non-functional system requirements are fixed.
- **Implementation of the application.** At this stage the program following the steps set out in the previous stage it is developed and implemented. In the development environment NetBeans a project with the necessary functions and objects is created, following the Java API that provides The Eye Tribe to get the expected solution.
- **Testing and conclusions.** At this stage the results of performing a series of tests in the application are presented. Then, the conclusions are clearly described and the advantages and disadvantages are explained.
- **Documentation.** This last stage represents the preparation of this document.

### I.3 Employed means

Next, a short list of the different means that have been necessary for the proper development of the work is going to be developed, explaining on each point of it what they are and its functionality within the project.

To carry out this Final Project, it has been decided to use **Microsoft Windows** as the operating system, specifically in its latest version, W10. It is based on windows and it became famous thanks to its graphical interfaces. Its latest version for desktop, at the time that this Project is being carried out, is Windows 10 (released in September 2014).

It has also been determined to use **Microsoft Word 2016**. It process texts created by Microsoft itself and today is the most commonly used when working with digital documents [2]. It was originally developed at IBM by the American programmer Richard Brodie in 1983.

#### The Eye Tribe EyeTracker

To carry out this Final Project, there is an eye tracking device offered by the company The Eye Tribe available. After a study, analysis and a comparison of the existing trackers on the current market (see Section 2.2.1), it has been concluded that this is the one which best fits the specifications required by the proposed solution.



ILLUSTRATION 3: EYE TRACKER OF THE EYE TRIBE

This device allows for making the eye-tracking analysis on a PC screen, tablet or smart phone. It has multiple applications such as:

- Market research and marketing.** By using heat maps (charts in which some specific areas of a web are highlighted by a color code based on a criteria of clicks and areas in which a pointer or the view pass through [3]) you can find out where users of websites or other applications pay more attention. Thus, a better distribution of the content and structure is achieved or, for example, you can place the most interesting products to sell in the most appropriate place.



ILLUSTRATION 4: EXAMPLE OF A HEAT MAP IN A WEBSITE

- Usability techniques.** This tracker consists of a very complete API which can encode a software that is able to use the view as a mouse, which, apart from providing convenience when handling devices, it is a way of allowing people with certain physical or psychological disabilities to use electronic devices. It can also be used to improve the user experience in virtual reality and videogames.





ILLUSTRATION 5: EXAMPLE OF USABILITY IN A TABLET

### **The Eye Tribe SDK 0.9.56**

The Eye Tribe company offers the SDK 0.9.56 tool in order to develop different utilities thanks to the very completed API that they have. Furthermore, in order to facilitate the task of the developer and so he can just focus on coding the application, it provides a software as an example, with the functions of calibration and evaluation of the eye tracking device with the user and the screen.

The main features of the Software Development Kit are [4]:

- It is compatible with Windows and OS X operating systems.
- It supports the programming languages C ++, C # and Java.
- Real-time flow of the coordinates of the gaze on the screen and pupil size.
- It consists of a developer manual with detailed information on the API, objects, functions and attributes available.

To develop this project a code that is responsible for performing authentication and synchronization of the machine with the user's gaze have to be programmed

### **JAVA programming language**

The algorithm that provides solution to the proposed problem in this Final Project is going to be implement using the programming language oriented to Java objects, since the EyeTribe Java SDK (name of API that The Eye Tribe has in Java) simplifies the communication with the EyeTribe server and facilitates the synchronization of the tracker with the eye.



ILUSTRATION 6: JAVA'S LOGOTYPE

Currently, it is one of the most widely used programming languages. Its syntax is largely derived from C # and C ++, but it has fewer low-level utilities than any of them.

Its main feature is that, due to its architecture, it is independent of the machine used. This is achieved because the code that it compiles it is running in virtual machines that are dependent on the platform (development environment in which it is running) [5].

### **Development Environment NETBEANS**

Netbeans is a free development and open code environment. It allows the use of a great number of development technologies for both desktop and web applications, or mobile devices. Among the technologies that it gives support Java, PHP, C #, C ++, HTML5, etc., can be found. Another important characteristic is that it can be installed in different operation systems like Linux, Mac OS X, Windows, etc. [6].



ILUSTRACIÓN 7: LOGOTIPO DE NETBEANS

Some of its main features are:

- Any update of the Java language is supported.
- It is a code editor with colored, versioning control, syntactical corrections and code templates.
- It has an specialised help assistant.

- Simplicity in project management thanks to its multiview platform and to the ability to structure the code in an organised way.
- Code optimization. The Profiler tries to optimize applications to run faster and with the minimal memory usage.
- Access to databases as Oracle and MySql and to application's servers such as Apache Tomcat, JBoss, GlassFish and WebLogic.
- The only requirement needed by the NetBeans development environment to run is to have Java JDK installed on your computer.

### Libraries: The Eye Tribe and Other

At this point, the libraries that will be used in the implementation of the authentication algorithm are going to be named and explained.

One of the libraries that has been used is the one that provides the development environment Eclipse, "JRE System Library". It consists of a large number of jar files, being only used a few of them as "jfxrt.jar" and "rt.jar".

It was also necessary to import the "eyetribe-java-0.6.60.jar" and "EyeTribeJavaFx.jar" external libraries, which have been acquired through an existing link on the developer page of The Eye Tribe company.

## I.4 Memory structure

This document describes in detail the execution of a study and implementation work of an authentication algorithm based on the eye-tracking technology, by selecting an image as a password, and whose development has been accomplished in a simulated environment using an oriented to objects programming language (Java).

The text consists of two large contents bloks that are differentiated by languages (the first one in English and the second one in Spanish).

- The first block, in **English**, consists of an introduction, an extended summary and a final conclusion.
- The second block, in **Spanish**, is the main one. It includes all the points that must be necessarily addressed in such documents (explained below).

Regarding the main block, it has seven chapters. Following, there is a brief description of what each one of them will address.



1. The first chapter consist of an **introduction**, in which a brief summary of the subject addressed in this project and some sub-sections are made: Motivation and objectives, development stages, used means and a memory structure.
2. In the second chapter the **problem statement** is presented, in which an analysis of the state of the art trying the different eye tracking authentication technologies and techniques is carried out. Besides, possible restrictions and the regulatory framework applicable to the subject are mentioned.
3. Then, in the third chapter, the **analysis and design of the technical solution** is performed.
4. After that, it proceeds to describe the **implementation** of the work. It shows that Java has been used as the programming language and the NetBeans tool as the development environment.
5. Once explained the previous points, **evaluation and results** of the developed authentication algorithm are detailed.
6. In the sixth chapter **project management** is determined, addresssing its planning and a budget analysis.
7. As a final point and closing the body of this text, **conclusions** are analyzed in a clear and transparent way, explaining its advantages and disadvantages. Finally, **future work lines** are examined, describing possible improvements to the algorithm.

In addition to these seven chapters, at the end, there are **references**.

## II. Extended abstract

### *Motivation*

Nowadays, with the huge increase and advance of technology, users are accessing online services constantly to carry out their everyday tasks. For this reason, there is an urgent need to seek for solutions and improvements in human-machine interactions.

At the same time, due to the fact that there is a huge number of services electronically available, the tendency of people is to keep all their files and sensitive data in their computers, mobile phones and tablets. Thus, an improvement in the security of these data is necessary, in order to reduce the likelihood that a cyberattacker can access to a device. One of the most important security mechanisms is authentication, which can be sometimes easily violated with attacks such as spoofing and keylogger.

As it has been shown in several studies, usability is essential when adopting and making a good use of a security technology, such as authentication is.

Because of all this, different investigations are being accomplished to find an authentication method that improves the security and usability features of traditional techniques, such as the authentication based on one factor. Among these improvements, we can highlight some techniques such as those based on two authentication factors, biometrics and eye-tracking; the latter being the newest and the one that is giving the most satisfactory results.

The eye-tracking technology is based on a heat tracking system that is able to calculate the exact place where a person is looking at, extracting information from someone's face and eyes. It has a big variety of applications, as it can be the replacement of the mouse by the user gaze, (utility that is very beneficial for disabled people), the creation of "heat maps", and also the authentication.

### *Goals and Methodology*

Due to the innovative technology underlying eyetracking and to the great usability and security benefits that it can bring to authentication techniques (providing solutions to the problems presented above), in this Final Project it has been decided to carry out the study and implementation of an authentication algorithm. The goal is to understand the implementation process, its difficulty, the benefits that the final assesment will present, and the levels of usability and security achieved.

In this paper, different **eye-tracking mechanisms** will be studied in order to understand them and to be able to decide which one is best in solving the proposed problem. Among the three methods of eye tracking (invasive, non-invasive and electrical potentials), the non-invasive type has been used. This method is the one that does not require contact with the user.

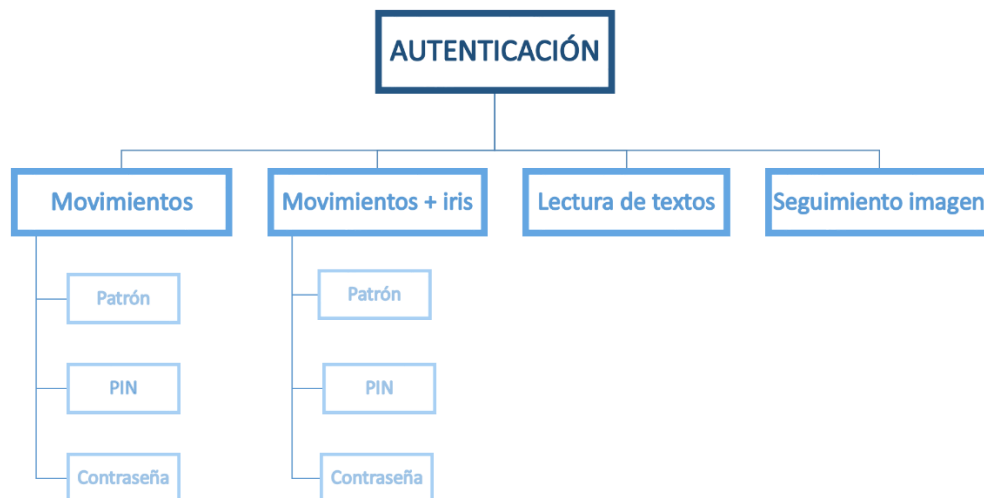
After that, the most competitive eye-tracking products that exist in the current market have been analysed, obtaining its technical and functional specifications. In this way, through a comparison of their characteristics, we can choose the one that best fits what we are looking for. When observing that the

product offered by the company The Eye Tribe has a significantly lower price than others (90 €), and that it is smaller in terms of size, it has been determined that the study will be applied to this device.

| <b>DEVICE</b>           | <b>Sampling rate</b> | <b>Accuracy</b> | <b>Tracking area</b> | <b>Dimensions</b> |
|-------------------------|----------------------|-----------------|----------------------|-------------------|
| The Eye Tribe [7]       | <b>VV</b>            | <b>VV</b>       | <b>VV</b>            | <b>X</b>          |
| SentiGaze [8]           | <b>V</b>             | <b>-</b>        | <b>V</b>             | <b>-</b>          |
| AEye Eye Tracker [9]    | <b>VV</b>            | <b>X</b>        | <b>-</b>             | <b>VV</b>         |
| Tobii Eye Tracking [10] | <b>V</b>             | <b>V</b>        | <b>-</b>             | <b>X</b>          |
| Eyegaze [11]            | <b>V</b>             | <b>V</b>        | <b>X</b>             | <b>V</b>          |

**TABLE 1: COMPARATIVE OF TECHNICAL FEATURES OF EVERY EYE-TRACKING DEVICE**

Finally, a study about the different authentication techniques that can be accomplished with an eye-tracker has been carried out. Based on this assessment, we decide which technique is going to be used in the project.



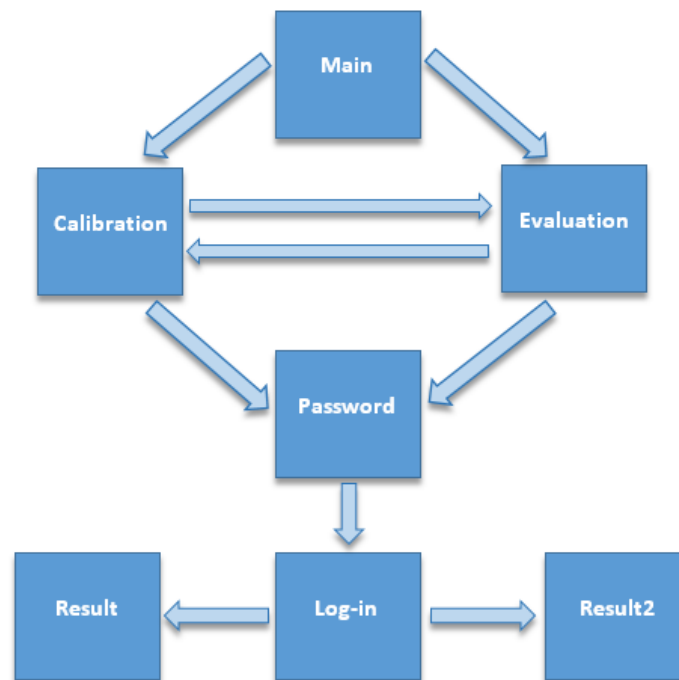
**ILLUSTRATION 8: DIAGRAM OF THE DIFFERENT AUTHENTICATION TECHNIQUES**

In conclusion, after analyzing and comparing different eyetracking technologies, the different products on the market and the various authentication techniques, it has been determined to solve the issue exposed in this Final Project by developing an algorithm that is based on eye movement in front of an image using the tracker offered by the Eye Tribe (non-invasive device). In the developed application it has sought to maintain a balance between security and ease of use, that is, using a sufficient number of images to choose the key one among them, so that the solution has a high level of security but not too high it diminishes usability by requiring a user login process that lasts too long.

In order to carry out the proper development of the solution, it has been required to employ various technological means, software and hardware. On the one hand, a computer with the Microsoft Windows operating system on its Windows 10 version has been used, as well as the word processor Microsoft Windows 2016 (offered free by the Carlos III University of Madrid). On the other hand, the core technological means that have been used are The Eye Tribe eye-tracker and its SDK, both obtained when acquiring the product.

To program the application it has been decided to use the Java programming language, since the EyeTribe Java SDK consists of a very simple and complete API; and the development environment NetBeans, to enable the use of a large number of development technologies.

En cuanto a la arquitectura que presenta la aplicación de autenticación que se propone, se ha decidido que el diseño más optivo es con 7 interfaces diferentes conectadas entre sí (como se muestra en la ilustración 9).



**ILUSTRATION 9: APPLICATION ARCHITECTURE**

When developing the user interface of the application, it is necessary to introduce several windows that allow the user to perform the log in operation. Besides, in order to achieve a better accuracy in the authentication operation, a calibration algorithm has been included. This algorithm calibrates the tracker according to the distance between the user and the device, the distance between his eyes and his position.

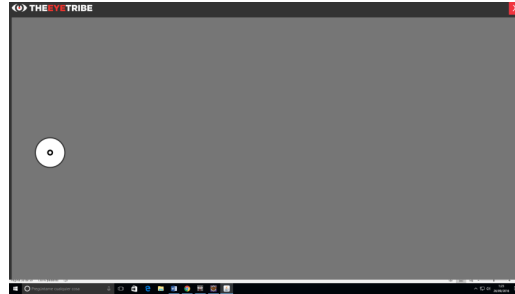


ILLUSTRATION 10: CALIBRATION WINDOW

It has also established a main window that appears when the platform starts. This has an animation that shows the position of the user's eyes seen from the tracker. So, the user knows whether he is too close or too far from the device, if his head is straight or twisted, or if he needs to be more focused.

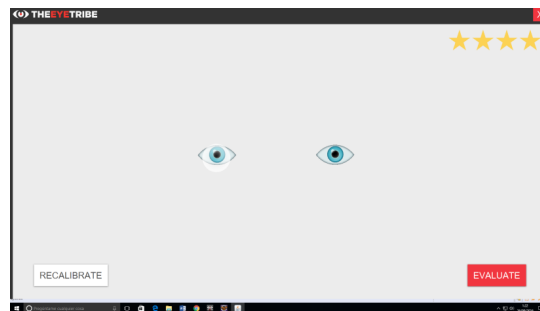


ILLUSTRATION 11: MAIN WINDOW

Furthermore, an evaluation window where the user is able to see if the calibration that has been carried out is successful or not, has been made. From that point, the user has the possibility to access directly to calibration (if there are any mistakes in this calibration and the user wants to repeat it) or to start choosing a password.

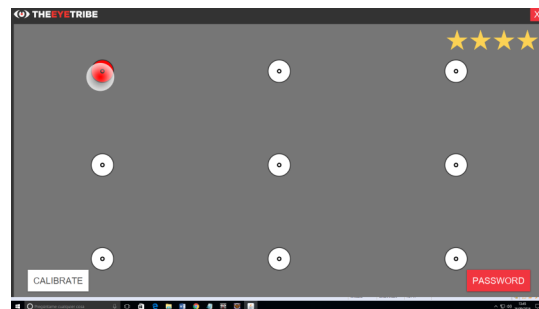
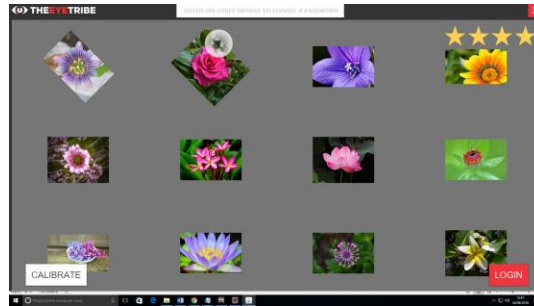


ILLUSTRATION 12: EVALUATION WINDOW

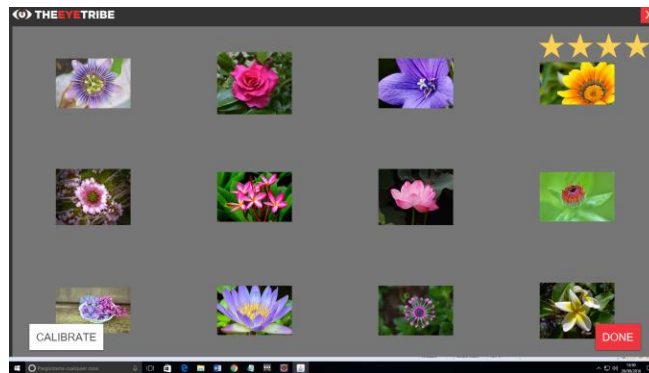
As an example, a window in which the user has to choose a password to authenticate himself has been added. Several images are displayed on the screen and the user has to fix his gaze on the three images he wants to save as a password. Once this is made, he has to press the "Done" button that directs him to the login page. This feature has been introduced in the application to make test easily. In a real case the user could only access to this page if it would be the first time he set up the password or if he would want to change it.





**ILLUSTRATION 13: PASSWORD WINDOW**

In the login window, the authentication algorithm based on eye movement to choose some key images by the user has been introduced. In every moment, it has been taken into account that the number of images displayed in the window has to be high enough so the algorithm provides a strong security, but not so high that it is no longer usable.



**ILLUSTRATION 14: LOG-IN WINDOW**

The last two application windows are used to show the user a message asking whether the password he has introduced is the right one or it is not.

Most of the implemented code has been defined by the author of this work. But some classes such as those responsible for configuring the server, contacting with it and receiving data, have been integrated thanks to a project (Java) that the company The Eye Tribe makes available to all its customers and developers on the web page GitHub. This is done so that all of them can focus on creating applications and utilities for its eye-tracking device.

Once the algorithm has been implemented, two types of tests have been carried out. In the first one, it has been observed if the application could be used with transparent glasses, obtaining a positive result. The second test consisted in checking whether the maximum operating distance was the one specified in the technical characteristics of the tracker (75 cm) and also finding out the minimum. It has also been found that the tool works properly at this distance and it has been observed that it can also log in with a bigger distance (up to 90 cm), but it shows some difficulties. In relation to the minimum distance its been set in 45 cm.



## III. Conclusions

### III.1 Conclusions

This Final Project, intended to study and implement an authentication algorithm based on the eye-tracking technology and to create a platform for login. This new technology is innovative and provides a great usability and security to the login techniques and, that's why it was chosen. This was proposed as a solution because the traditional techniques did not respond to current needs for security and ease of use.

In addition, the authentication method based on the eyes movement of an individual in front of some images (placed ahead of the tracker device) was chosen. This was decided by observing a comparison of some current innovative techniques. This comparison showed that, apart from providing a **greater security and usability** than the others, it was the newest.

To make the application design, it was first analysed the system requirements, evaluating how many windows were needed and how they had to be. After that, the classes and methods that every of them would require were studied, with the aim of finding the optimal architecture.

Taken into account the programming skills of the author of this project and that the SDK Java API was the easiest one of those offered by The Eye Tribe, it was decided to use the programming language oriented to Java objects.

Once the work using the codified design was developed and the appropriate tests performed, it was observed that the application worked properly fulfilling all the specifications previously required.

Moreover, when the project has been finished and the technology has been known in depth, it has been possible to observe that there are some changes that would improve the proposed solution.

### III.2 Future work lines

As seen in the previous point, the developed application meets all the objectives set.

However, when enough knowledge about the technology, the device, the development environment and the programming language have been acquired, it is observed that there are new features or improvements that could be implemented in the future. Here there are some of them:

The first and most important improvement is to increase the **security** of the authentication technique developed by using a greater number of images on the authentication page. Thus, the odds of matching the images chosen as the password for the user would be reduced. Another option to improve security is to use the authentication method developed in this project more than once and then to compare if the

chosen images were saved as passwords. These two options are compatible with each other and together, they will significantly decrease the likelihood of intrusion.

Another future improvement could be the implementation of this application to other operative systems compatible with SDK as Mac OS X. Besides, different versions for mobile devices as smartphones or tablets could be accomplished, in case that they have installed the Android operating system.

With the aim of confirm that the chosen authentication technique is best one, an algorithm based on one of them could be implemented. In this way, the results regarding usability and security that they provide could be compared.

Finally, the English language has been used to generalise the application. A future step could be the increasing of languages, implementing the application, for instance, in French and Spanish in order to make it accessible to a higher number of people.

## Capítulo 1: Introducción

Durante las últimas décadas las tecnologías han experimentado un gran avance. Esto se debe, sobre todo, a la fuerte expansión de internet, provocando que cada vez sea mayor el número de servicios a los que los usuarios pueden acceder de forma electrónica.

En la actualidad se puede escuchar música, ver la televisión, realizar gestiones administrativas, comunicarse con otros individuos, comprar todo tipo de necesidades como ropa y comida, jugar y realizar otras muchas funciones únicamente a través de internet. Debido a esta amplia lista de servicios disponibles, surge la necesidad de buscar soluciones y mejoras en la interacción persona-máquina.

Una nueva tecnología que puede hacerlo es la basada en el seguimiento de ojos (en inglés “*eye-tracking*”) a través de la cual se consigue que la experiencia de usuario y accesibilidad seas únicas permitiendo utilizar la vista como si fuese un ratón de ordenador, algo que especialmente beneficiaría a personas con ciertas discapacidades físicas o psíquicas que, finalmente, serían capaces de realizar tareas telemáticas por sí solas. Esta técnica también es capaz de mejorar la arquitectura de páginas web a través del estudio de “*heat maps*”, realizar estudios de marketing digital e incluso crear aplicaciones que aporten seguridad a los dispositivos electrónicos.

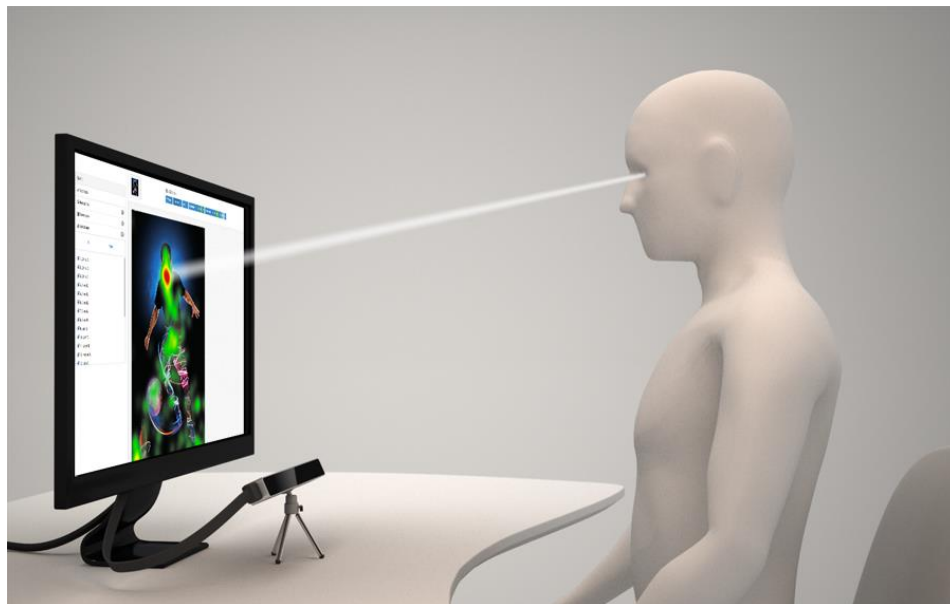


ILUSTRACIÓN 15: INTERACCIÓN PERSONA-MÁQUINA CON LA TECNOLOGÍA EYETRACKING

## 1.1 Motivación y objetivos

### MOTIVACIÓN

Hoy en día, las personas guardan todos sus archivos y datos sensibles en sus ordenadores, móviles y tabletas. Debido a esto, se requiere de una fuerte seguridad que impida a terceras personas penetrar en los dispositivos pudiendo así llevarse información o infectar el terminal.

Con el gran aumento de delitos de este tipo que hay en este momento, se está buscando mejorar la seguridad con el fin de disminuir las probabilidades de que un delincuente pueda acceder a un dispositivo. En especial, se pretende mejorar la primera y más importante barrera de seguridad, la autenticación.

Paralelamente, se debe tener en cuenta que, como dijo U. Jendricke en un artículo [1] y posteriormente demostró, la usabilidad siempre ha sido y siempre será un aspecto crucial en la adopción y buen uso de tecnologías de seguridad.

Actualmente, se han propuesto varias mejoras a la autenticación tradicional basada en un único factor, es decir, aquella en la cual un elemento es conocido por el usuario (contraseña o patrón). De entre estas mejoras destacan:

- Las basadas en **dos factores de autenticación**, las cuales son una combinación de dos métodos de autenticación distintos en las que existe un elemento que el usuario sabe (contraseña o patrón) y un elemento que el usuario posee (DNI y número de teléfono).
- Las basadas en la **biometría** que consisten en la verificación de la identidad de una persona teniendo en cuenta componentes morfológicos que son inherentes y que únicamente se dan en ese sujeto. Es decir, rasgos como puede ser la huella dactilar, el iris y la voz.
- Las basadas en una tecnología llamada **Eyetracking**, la cual es capaz de calcular el lugar exacto que una persona está mirando a través de información extraída de su cara y ojos.

Se ha probado que las diferentes técnicas de autenticación basadas en aspectos biométricos y en la tecnología eyetracking, aportan una mayor usabilidad al usuario frente a las utilizadas tradicionalmente.

De entre estas técnicas que presentan mejoras, destaca la basada en eyetracking al ser la más innovadora. Esta tecnología permite implementar diferentes métodos con los que un usuario se puede autenticar: dibujando patrones con la vista, leyendo textos, siguiendo con la mirada una o varias imágenes, etc. Dichos métodos (detallados en la sección 2.1.2) son capaces de aportar una fuerte seguridad, además de no ser complicados de recordar y ser accesibles incluso para personas con ciertas discapacidades, mientras que con las técnicas de autenticación tradicionales necesitarían de algún movimiento físico, ya sea manejando un ratón o el dedo.

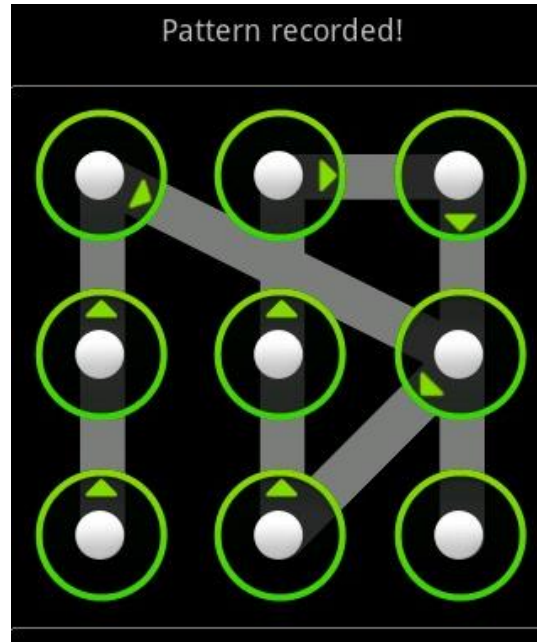


ILUSTRACIÓN 16: EJEMPLO DE AUTENTICACIÓN MEDIANTE EL DIBUJO DE UN PATRÓN

Debido a la gran usabilidad y seguridad que puede aportar la tecnología eyetracking en las técnicas de autenticación, se va a diseñar e implementar un algoritmo de autenticación con el fin de comprender el proceso de implementación, las prestaciones que presenta la aplicación final y la facilidad de uso y seguridad que es capaz de proporcionar.

## OBJETIVOS

El objetivo general del trabajo es estudiar, implementar y evaluar un prototipo de bajo coste para autenticación basada en la tecnología eyetracking con el fin de conseguir mejorar la usabilidad y seguridad en la autenticación de un usuario.

Se va a utilizar un dispositivo de seguimiento de ojos, tracker, que permite capturar y registrar el punto que una persona está mirando delante de una pantalla. Con los datos que devuelve el tracker se pretende ser capaz de autenticar a un sujeto mediante una técnica que se elegirá durante el Proyecto.

Se tratará de buscar un método de autenticación que sea innovador, que todavía no se haya implementado o probado, que sea fácil de desarrollar y que aporte características esenciales a la autenticación como facilidad de uso y seguridad.

Para poder alcanzar el objetivo general y la conclusión de trabajo, se han de definir una serie de subobjetivos a seguir. Estos subobjetivos son los siguientes:

1. **Analizar y comparar las distintas tecnologías de eyetracking.** Se realizará un estudio de las diferentes tecnologías de eyetracking existentes en el mercado como The Eye Tribe, SentiGaze, Tobii, etc. Posteriormente, se compararán teniendo en cuenta las características técnicas y

funcionales de cada una y finalmente, se seleccionará la más adecuada para la solución del trabajo.

2. **Estudiar las distintas técnicas de autenticación.** Se llevará a cabo un análisis de las diversas técnicas de autenticación que pueden encajar como solución al problema y se elegirá aquella que sea innovadora, fácil de desarrollar y que aporte una gran usabilidad y seguridad.
3. **Familiarizarse con el tracker.** Una vez conocidas sus funcionalidades y sus características técnicas, se probarán todas ellas en el dispositivo físico hasta examinar cada opción que ofrezca y se evaluarán sus limitaciones.
4. **Estudiar las librerías existentes.** Vistas las posibilidades del dispositivo hardware, es necesario estudiar y comprender cada una de las funciones, métodos y objetos disponibles en el API. Se verá si entre esas librerías existen los elementos que se requieren para llevar a cabo el desarrollo de la aplicación, lectura de los ojos, posición de los mismos, representación de imágenes o puntos, etc. También será imprescindible analizar los diferentes lenguajes de programación disponibles en el SDK para así, en función de cuál sea el más completo y sencillo, elegir uno de ellos para implementar el algoritmo.
5. **Adaptarse al lenguaje de programación y al entorno de desarrollo.** Cuando ya se ha decidido el lenguaje de programación, el siguiente paso que hay que dar es formarse en el lenguaje y escoger un entorno de desarrollo que lo soporte.
6. **Diseñar e implementar una arquitectura de autenticación.** Una vez se ha elegido la técnica de autenticación y el lenguaje de programación más adecuados y se ha estudiado el API, es el momento de diseñar la arquitectura que tendrá la aplicación de autenticación. Se buscará que esta sea sencilla de implementar, fácil de entender por el usuario y lo más visual posible. Posteriormente, se procederá a realizar la implementación de la arquitectura en el entorno de desarrollo seleccionado anteriormente.
7. **Evaluar las prestaciones de la solución implementada frente a mecanismos tradicionales.** Cuando ya se haya desarrollado el algoritmo, se realizarán diversas pruebas con el fin de evaluar sus prestaciones y limitaciones. Después, se comparará, en cuanto a prestaciones, la solución obtenida frente a los mecanismos de autenticación tradicionales. Sobre todo, se analizarán los aspectos que tienen que ver con la usabilidad y seguridad del algoritmo y finalmente, se sacará una conclusión final.

## 1.2 Fases de desarrollo

Para el desarrollo de este Trabajo de Fin de Grado se van a llevar a cabo una serie de etapas perfectamente diferenciadas, con el fin de cumplir los objetivos fijados:



- **Análisis de la tecnología.** En esta fase, primero se elabora la planificación inicial del proyecto y posteriormente, se realiza un amplio análisis del entorno que se desea estudiar, las tecnologías y herramientas que se utilizarán. Para ello, se examina más en profundidad la documentación proporcionada por The Eye Tribe sobre el dispositivo de seguimiento de ojos y se comparan las diferentes técnicas de autenticación.
- **Diseño de la solución.** En esta segunda fase se definen las etapas que se seguirán en la implementación de la solución. Además, se toman decisiones de diseño, como la técnica de autenticación y el lenguaje de programación que se van a emplear, y se fijan los requisitos funcionales y no funcionales del sistema
- **Implementación de la aplicación.** En esta etapa se desarrolla e implementa el programa siguiendo las etapas fijadas en la fase anterior. En el entorno desarrollo NetBeans se crea un proyecto con las funciones y objetos necesarios, siguiendo el API de Java que proporciona The Eye Tribe, para conseguir la solución esperada.
- **Pruebas y conclusiones.** En esta etapa se presentan los resultados obtenidos de realizar una serie de pruebas en la aplicación. A continuación, se exponen las conclusiones obtenidas de forma clara y se explican las ventajas y desventajas.
- **Documentación.** Esta última etapa representa la elaboración de este documento.

### 1.3 Medios empleados

A continuación, se va a elaborar una breve lista de los diferentes medios que ha sido necesario emplear para el correcto desarrollo del trabajo, explicando en cada punto de la misma en qué consiste y su funcionalidad dentro del proyecto.

Para llevar a cabo este Trabajo de Fin de Grado, se ha decidido utilizar como sistema operativo **Microsoft Windows**, concretamente en su última versión, W10. Está basado en ventanas y se hizo famoso gracias a sus interfaces gráficas. Su versión más reciente para equipos de escritorio, en el momento de realización de esta memoria, es Windows 10, lanzada en septiembre de 2014.

También se ha determinado emplear **Microsoft Word 2016**. Se encarga de procesar textos creados por el propio Microsoft y, actualmente, es de los más utilizados a la hora de trabajar con documentos digitales [2]. Originalmente, fue desarrollado en IBM por el programador estadounidense Richard Brodie en el año 1983.

### 1.3.1 Eyetracker de The Eye Tribe

Para la realización de este TFG, se dispone de un dispositivo de seguimiento de ojos que ofrece la compañía *The Eye Tribe*. Tras un estudio, análisis y comparación de los trackers existentes en el mercado actual (ver Sección 2.2.1), se ha llegado a la conclusión de que este es el que más se adapta a las especificaciones que requería la solución propuesta.



ILUSTRACIÓN 17: EYE TRACKER DE THE EYE TRIBE

Este dispositivo permite realizar análisis de seguimiento de ojos sobre una pantalla de PC, tableta o móvil inteligente. Tiene múltiples aplicaciones como:

- Estudios de **mercado y marketing**. Mediante el uso de *heat maps* (gráficos en los que se resaltan por medio de un código de colores zonas concretas de una web en base a criterios de clics y áreas por las que se pasa con más frecuencia un puntero o la vista [3]) se puede averiguar dónde ponen más atención los usuarios de páginas webs u otras aplicaciones. De esta forma se consigue una mejor distribución del contenido y estructura o, por ejemplo, poder colocar los productos más interesantes o más necesarios de vender en el lugar más indicado.



ILUSTRACIÓN 18: EJEMPLO DE UN HEAT MAP EN UNA PÁGINA WEB

- Técnicas de **usabilidad**. Este tracker consta de un API muy completo con el que se puede codificar un *software* capaz de utilizar la vista a modo de ratón, lo que, aparte de aportar comodidad a la hora de manejar los dispositivos, es una forma de permitir a personas con ciertas discapacidades físicas o psicológicas utilizar aparatos electrónicos. También se puede emplear para mejorar la experiencia de usuario en la realidad virtual y en juegos.



ILUSTRACIÓN 19: EJEMPLO DE USABILIDAD EN UNA TABLETA

### 1.3.2 The Eye Tribe SDK 0.9.56

La empresa *The Eye Tribe* ofrece la herramienta The Eye Tribe SDK 0.9.56 para poder, con el completo API que presentan, desarrollar diferentes utilidades. Además, con el fin de facilitar la tarea del

desarrollador y que se pueda centrar en codificar la aplicación, proporciona un software a modo de ejemplo, con las funciones de calibración y evaluación del dispositivo de seguimiento de ojos con el usuario y la pantalla.

El paquete SDK que se obtiene al comprar la licencia consta de un servidor y una aplicación de calibración. El servidor gestiona la comunicación con el dispositivo lector de ojos y, a través de algunos parámetros, lo configura. Por otro lado, la interfaz de calibrado solo se puede utilizar para realizar las primeras pruebas. En una aplicación con funcionalidad, se tendría que crear el calibrado desde cero.

Las características principales del *Software Development Kit* son [4]:

- Es compatible con los sistemas operativos Windows y OS X.
- Soporta los lenguajes de programación C++, C# y Java.
- Flujo a tiempo real de las coordenadas de la mirada en la pantalla y el tamaño de las pupilas.
- Consta de un manual de desarrollador con la información detallada relativa al API, los objetos, funciones y propiedades disponibles.

Para desarrollar el trabajo es necesario programar un código que se encargue de realizar la autenticación y la sincronización de la máquina con la mirada del usuario.

### 1.3.3. Lenguaje de programación JAVA

El algoritmo que da solución al problema propuesto en este TFG se ha decidido implementar utilizando el lenguaje de programación orientado a objetos Java ya que, el EyeTribe Java SDK (denominación del API que tiene The Eye Tribe en Java) simplifica la comunicación con el servidor EyeTribe y facilita la sincronización del tracker con el ojo.



ILUSTRACIÓN 20: LOGOTIPO DE JAVA

Actualmente, es uno de los lenguajes de programación más utilizados. Su sintaxis deriva en gran parte de C# y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos.

Su principal característica es que, debido a su arquitectura, es independiente de la máquina en la que se esté utilizando. Esto se consigue ya que el código que compila se ejecuta en máquinas virtuales que sí son dependientes de la plataforma (entorno de desarrollo en el que se esté corriendo) [5].

### 1.3.4 Entorno de desarrollo NETBEANS

Netbeans es un entorno de desarrollo gratuito y de código abierto. Permite utilizar una gran cantidad de tecnologías de desarrollo tanto para escritorio como aplicaciones web o para dispositivos móviles. Entre las tecnologías a las que da soporte se encuentran Java, PHP, C#, C++, HTML5, etc. Otra cualidad importante es que puede ser instalada en diferentes sistemas operativos como Linux, Mac OS X, Windows, etc [6].



ILUSTRACIÓN 21: LOGOTIPO DE NETBEANS

Algunas de sus características principales son:

- Cualquier actualización del lenguaje Java es soportada.
- Editor de código con coloreado, control de versiones, correcciones sintácticas y plantillas de código.
- Asistente de ayuda especializado.
- Simplicidad en la gestión de proyectos gracias a su plataforma multivista y a la capacidad de estructurar el código de manera ordenada.
- Optimización del código. El Profiler intenta optimizar las aplicaciones para que se ejecuten más rápido y con el mínimo uso de memoria.
- Acceso a base de datos como Oracle y MySQL y a servidores de aplicaciones como Apache Tomcat, JBoss, GlassFish y WebLogic.

El único requisito que necesita el entorno de desarrollo NetBeans para ejecutarse es tener instalado en el equipo un JDK de Java.

### 1.3.4 Librerías: The Eye Tribe y Otras

En este punto se van a nombrar y explicar las librerías que se utilizarán en la implementación del algoritmo de autenticación.

Una de las librerías que se ha utilizado es la que proporciona el entorno de desarrollo Eclipse, “**JRE System Library**”. Está formada por un gran número de ficheros *jar*, de los cuales solo se han utilizado unos pocos como “*jfxrt.jar*” y “*rt.jar*”.

También ha sido necesario importar las librerías externas “*eyetribe-java-0.6.60.jar*” y “*EyeTribeJavaFx.jar*”, las cuales se han adquirido a través de un enlace existente en la página de desarrolladores de la empresa *The Eye Tribe*.

### 1.4. Estructura de la memoria

El presente documento describe detalladamente la realización de un trabajo de estudio e implementación de un algoritmo de autenticación basado en la tecnología *eye-tracking* mediante la selección de una imagen como contraseña, cuyo desarrollo ha sido llevado a cabo en un entorno simulado utilizando un lenguaje de programación orientado a objetos (Java).

La memoria consta de dos grandes bloques diferenciados por el idioma, la primera en inglés y la segunda en español.

- El primer bloque, en **inglés**, está constituido por una introducción, un resumen extendido y una conclusión como final.
- El segundo bloque, en **español**, es el principal ya que comprende todos los puntos que necesariamente deben ser tratados en este tipo de documentos (explicados a continuación).

En cuanto al bloque principal, está compuesto por **siete capítulos**. A continuación, se encuentra una breve descripción de qué trata cada uno de ellos.

1. El primer capítulo consta de una **introducción** en la que se realiza un breve resumen del tema tratado en el trabajo y unos sub-apartados: Motivación y objetivos, Fases de desarrollo, Medios empleados y Estructura de la memoria.
2. En el segundo capítulo se presenta el **planteamiento del problema**, en el cual, se lleva a cabo un análisis del estado del arte tratando las diferentes tecnologías y técnicas de autenticación de seguimiento de ojos. Además, se mencionan las posibles restricciones y el marco regulador que aplican al tema tratado.



3. A continuación, en el tercer capítulo, se encuentra el **análisis y diseño de la solución técnica**. Primero se especifican los requisitos que debe cumplir la aplicación y después, se trata la arquitectura de la misma.
4. Tras ello, se procede a describir la **implementación** del trabajo. Indicando que se ha utilizado como lenguaje de programación *Java* y como entorno de desarrollo la herramienta *NetBeans*.
5. Una vez explicados los puntos anteriores, se detalla la **evaluación y resultados** del algoritmo de autenticación elaborado.
6. En el sexto capítulo se define la **gestión del proyecto**, tratando la planificación del mismo, así como un análisis presupuestario.
7. Como último punto y cierre del cuerpo de la memoria, se analizan las **conclusiones** de forma clara y transparente, explicando sus ventajas y desventajas, y se examinan las **líneas futuras**, describiendo posibles mejoras del algoritmo.

Además de estos siete capítulos, al final de los mismos, se encuentran las **referencias**.





## Capítulo 2: Planteamiento del problema

### 2.1 Análisis del estado del arte

#### 2.1.1 Tecnologías de seguimiento de ojos o *Eyetracking*

**Eyetracking** es un sistema de seguimiento del movimiento ocular capaz de calcular, mediante sensores, el lugar exacto que una persona está mirando a través de información extraída de su cara y ojos. Se basa en la iluminación del ojo con infrarrojos y la utilización de algoritmos matemáticos complejos.

Esta tecnología se puede utilizar en una amplia variedad de aplicaciones que se dividen en dos categorías, pasivas y activas. Las activas implican el control de dispositivos como, por ejemplo, mover el cursor de un ordenador con los ojos, manejar los controles de un videojuego con la mirada y escribir. En cambio, las aplicaciones pasivas realizan funciones como, por ejemplo, analizar el rendimiento del diseño y el posicionamiento de la publicidad de una página web.

#### *Historia*

En el año 1800 se comenzaron a realizar estudios basándose en los movimientos oculares mediante la observación directa de un individuo. En 1879, el doctor Louis Emile Javal, parisino nacido en 1839, se dio cuenta de que el barrido de los ojos a lo largo del texto no marca la lectura del mismo, como se creía anteriormente, sino las paradas que realiza el lector [12]. A partir de este momento se hicieron varias preguntas, ¿Cuántas veces se detiene el lector cuando lee un texto?, ¿dónde lo hace?, ¿qué patrones siguen esas detecciones?, ¿en qué palabras lo hace?

Más tarde, el científico Edmund Huey [7] fabricó el primer prototipo de un dispositivo seguidor de ojos con una lente que se colocaba en los ojos del sujeto a seguir. El objetivo estaba conectado a un puntero de aluminio que se movía en respuesta al movimiento del ojo.

Guy Thomas Buswell construyó el primer dispositivo no intrusivo. Se basó en el uso de haces que se reflejan en el ojo para grabarlos en una película. No solo se centró en la lectura de textos, sino que también llevó a cabo estudios sobre la visualización de imágenes.

A partir de 1970, las investigaciones sobre el seguimiento ocular se hicieron más importantes, invirtiéndose más recursos. El 1980, investigadores estudiaron cómo los usuarios podían realizar búsquedas en los menús del ordenador. También utilizaron el seguimiento de los ojos en tiempo real para ayudar a personas con discapacidades y para estudiar la interacción de los usuarios con interfaces de ordenadores y así plantear cambios en el diseño. Otra área de investigación reciente se centra en el desarrollo Web.

Actualmente, esta tecnología está adquiriendo una mayor atención por parte de empresas gracias al aumento de soluciones comerciales y a la fuerte disminución de precio de las mismas.

## *Tipos de seguimiento de ojos*

### **Invasivo**

Se utilizan lentes especiales, con un espejo o un sensor de campo magnético, que se adhieren al ojo del sujeto. Las mediciones que se han realizado con esta técnica han aportado grabaciones muy detalladas de los movimientos oculares [14].

### **No invasivo**

En este tipo de seguimiento no es necesario el contacto. La luz infrarroja se refleja en los ojos y se captura utilizando un sensor óptico. La información recogida se analiza para extraer la rotación de los ojos y los cambios en los reflejos. Los métodos ópticos basados en la grabación de video son utilizados para el seguimiento de la mirada y están bien considerados al no ser invasivos y el coste es bajo. Se utiliza un dispositivo externo específico para calcular las coordenadas (x e y) a las que el usuario está mirando en la pantalla [15].

### **Potenciales eléctricos**

Se detecta el movimiento de los ojos a través de electrodos colocados a su alrededor. Esto se debe a que los ojos son un campo de potencial eléctrico constante. Se puede modelar para generar un dipolo con el polo positivo en la córnea y el polo negativo en la retina [16].

## *Aplicabilidad*

Las técnicas de eye-tracking tienen diversas aplicaciones en una gran variedad de áreas de estudio como puede ser el marketing, la publicidad, la medicina, etc, pudiéndose utilizar incluso para mejorar la accesibilidad en discapacitados [9].

Las aplicaciones más importantes son [9]:

- **Publicidad online**

A través de esta tecnología se pueden conocer los lugares en los que los usuarios ponen mayor atención, consiguiendo así una información que puede utilizarse a la hora de estructurar la publicidad de la mejor forma posible.

- **Diseño web y de software**

Estudiando las zonas donde el usuario pone mayor atención en webs e interfaces, los desarrolladores son capaces de determinar el lugar más adecuado para cualquier elemento. Esto se consigue observando la información obtenida mediante su representación en *heat maps*.

- **Medicina**

Se están llevando a cabo investigaciones para fabricar unas gafas, con esta tecnología, capaces de mostrar en 3D lo que el profesional esté mirando.

- **Accesibilidad para personas discapacitadas**

Con los productos de eye-tracking, los discapacitados conseguirán grandes mejoras. Se ha conseguido asociar el ratón de un ordenador con la mirada de un sujeto.

- **Juegos**

Hoy en día, la industria de los videojuegos es muy importante. Con la tecnología de seguimiento ocular se podrá conseguir una mejor experiencia de usuario dejando atrás la necesidad de utilizar las manos para manejar los paneles.

- **Autenticación**

La utilización del seguimiento de ojo para la autenticación aporta un gran nivel de seguridad al dispositivo.

### *Productos de eye-tracking*

En este apartado se analizarán los productos más competitivos en la actualidad que emplean esta tecnología con el fin de obtener sus especificaciones y así poder elegir el que más se adecúe a lo que se busca. Todas las especificaciones que se utilizarán para compararlos estarán basadas en la información disponible de forma pública.

De entre los tres métodos de seguimiento de ojos, este trabajo se centrará en el tipo no invasivo.

### **THE EYE TRIBE**

Este tracker ha sido desarrollado por The Eye Tribe, una empresa danesa fundada hace tan solo 4 años que, desde sus comienzos, se ha centrado en esta tecnología, consiguiendo hacerse un hueco entre las más competitivas. El primer prototipo fue mostrado en el año 2013 en una tableta con sistema operativo Windows [17].



ILUSTRACIÓN 22: LOGOTIPO THE EYE TRIBE

#### **Características funcionales**

- Funciona mejor en lugares cerrados sin que la luz solar de directamente al dispositivo.
- Posibilidad de llevar lentillas y gafas.

#### **Características técnicas**

|                               |                                 |
|-------------------------------|---------------------------------|
| <b>Frecuencia de muestreo</b> | 30Hz a 75Hz                     |
| <b>Rango de precisión</b>     | 0.5° – 1°                       |
| <b>Calibrado</b>              | 6, 9 o 12 puntos                |
| <b>Área de Tracking</b>       | 50cm x 30cm a 65cm de distancia |

|                     |                 |
|---------------------|-----------------|
| Tamaño de pantalla  | Inferior a 27"  |
| API/SDK             | Java, C++ y C#  |
| Dimensiones (W/H/D) | 2 x 2 x 20 (cm) |
| Conexión            | USB3.0          |

TABLA 2: CARACTERÍSTICAS TÉCNICAS DEL TRACKER THE EYE TRIBE

\*Fuente: <https://theeyetribe.com/products/>



ILUSTRACIÓN 23: TRACKER DE THE EYE TRIBE

### Posibles aplicaciones

Como se puede ver en la página web de The Eye Tribe [16], las posibles aplicaciones que soporta este dispositivo son:

- Los buscadores Web y lectores pdf pueden desplazarse automáticamente mientras el usuario está leyendo.
- Hacer zoom en mapas mirando un icono.
- Abrir una ventana manteniendo la mirada en ésta.
- Jugar a diferentes juegos en primera persona como “shooters” donde el usuario apunta con la mirada y dispara con el ratón.
- Un teclado en la pantalla donde se pueden escribir textos, enviar correos, hablar en chats, etc. Esta aplicación es muy útil para personas con discapacidades motoras.
- Analizar el seguimiento de ojo de usuarios, para el posterior estudio de estos datos y mejorar diseños en páginas web, interfaces, etc, haciéndolas más atractivas.
- Métodos de inicio de sesión o *login*.

### SENTIGAZE

El dispositivo de seguimiento ocular *SentiGaze* ha sido desarrollado por la compañía Neurotechnology, fundada en 1990 en la capital lituana de Vilna. Esta empresa desarrolla algoritmos y software para productos biométricos como el reconocimiento de huellas dactilares, facial, ocular y de voz. En lanzó su primer producto que llevaba a cabo identificaciones a través de huellas dactilares y ahora proporciona soluciones completas con el hardware y software necesario a más de cien países [18].



ILUSTRACIÓN 24: LOGOTIPO NEUROTECHNOLOGY

### Características funcionales

- Funciona necesariamente con una *webcam* que cumpla unos requisitos técnicos específicos: resolución mayor o igual a 640 x 480 píxeles y accesible vía DirectShow.
- Únicamente es operable en PCs.
- Necesariamente la cámara debe estar situada debajo de la pantalla.

### Características técnicas

|                        |                             |
|------------------------|-----------------------------|
| Frecuencia de muestreo | 30 Hz                       |
| Rango de precisión     | No disponible               |
| Calibrado              | 9, 12 y 16 puntos           |
| Área de Tracking       | Específico de la webcam     |
| Tamaño de pantalla     | Inferior a 24"              |
| API/SDK                | Visual Basic .Net, C++ y C# |
| Dimensiones (W/H/D)    | Específicas de la webcam    |
| Conexión               | Específica de la webcam     |

TABLA 3: CARACTERÍSTICAS TÉCNICAS DEL TRACKER SENTIGAZE

\*Fuente: <http://www.neurotechnology.com/sentigaze-technical-specifications.html>

### Características técnicas PC

|            |  |
|------------|--|
| Procesador | x86 (32-bit) o x86-64 (64-bit)<br>3GHz o más |
| RAM        | Al menos 256MB                               |

TABLA 4: CARACTERÍSTICAS TÉCNICAS DEL PC

\*Fuente: <http://www.neurotechnology.com/sentigaze-system-requirements.html>

### Características técnicas webcam

|                          |  |
|--------------------------|--|
| <b>Resolución</b>        | Al menos 640 x 480 píxeles                 |
| <b>Sistema operativo</b> | Windows: XP, Vista, 7, 8, Server 20012,... |

**TABLA 5: CARACTERÍSTICAS TÉCNICAS WEB CAM**

\*Fuente: <http://www.neurotechnology.com/sentigaze-system-requirements.html>

### Posibles aplicaciones

- Generar “Heat maps” y otras estadísticas sobre múltiples materias. Útil para mejorar la usabilidad de páginas web y aplicaciones y para colocación de anuncios.
- Utilizar el ojo como un puntero.

### AEYE EYE TRACKER

EyeTech es una empresa de origen estadounidense situada en Mesa, Arizona. Fue fundada en 1996 y desde sus inicios se ha centrado en la creación de algoritmos, software y hardware especializado en sistemas de seguimiento de ojo. Ofrece una amplia variedad de productos de diferentes características técnicas, tamaños y precio, siendo el que más se adapta a nuestros requerimientos el llamado AEye Eye Tracker [19].



**ILUSTRACIÓN 25: LOGOTIPO EYETECH**

### Características funcionales

- Posibilidad de operar en portátiles, PCs, Smart phones, Raspberry Pi, etc.

### Características técnicas

|                               |                     |
|-------------------------------|---------------------|
| <b>Frecuencia de muestreo</b> | 40Hz a 200Hz        |
| <b>Rango de precisión</b>     | 1°                  |
| <b>Calibrado</b>              | No disponible       |
| <b>Área de Tracking</b>       | No disponible       |
| <b>Tamaño de pantalla</b>     | Cualquiera          |
| <b>API/SDK</b>                | C++ y C#            |
| <b>Dimensiones (W/H/D)</b>    | 2.5 x 2.7 x 29 (cm) |

**Conexión**

USB2.0 y GPIO

**TABLA 6: CARACTERÍSTICAS TÉCNICAS DEL TRACKER AEYE EYE**

\*Fuente: <http://www.eyetechds.com/board-level-oem.html>



**ILUSTRACIÓN 26: TRACKER DE EYE TECH**

**Posibles aplicaciones**

- Utilizar únicamente el ojo para interactuar con cualquier dispositivo (PCs, tabletas, etc).
- Analizar el seguimiento de ojo de usuarios, para el posterior estudio de estos datos y mejorar diseños en páginas web, interfaces, etc, haciéndolos más atractivos.

**TOBII EYE TRACKING**

El grupo Tobii se inició en el año 2001 en Suecia. En 2002, lanzaron su primer dispositivo tracker, el cual supuso un gran salto para la empresa al darse a conocer mundialmente. A partir de ese momento dividieron la compañía en cuatro grandes departamentos. Cada uno de ellos se encarga de desarrollar dispositivos específicos para aportar una solución: Tobii Dynavox, educativas; Tobii Pro, comportamiento humano; Tobii Tech, accesibilidad y Tobii EyeX, desarrollo de juegos para PC's, tabletas, etc [20].



**ILUSTRACIÓN 27: LOGOTIPO TOBII EXPERIENCE**

**Características funcionales**

- Posibilidad de operar en portátiles y PCs.
- Necesidad de Windows 10 como S.O.

**Características técnicas**

|                               |                  |
|-------------------------------|------------------|
| <b>Frecuencia de muestreo</b> | 60Hz             |
| <b>Rango de precisión</b>     | 0.45°            |
| <b>Calibrado</b>              | 6, 9 o 12 puntos |
| <b>Área de Tracking</b>       | No disponible    |

|                     |                   |
|---------------------|-------------------|
| Tamaño de pantalla  | Inferior a 27"    |
| API/SDK             | .Net, C++         |
| Dimensiones (W/H/D) | 2 x 1.5 x 23 (cm) |
| Conexión            | USB               |

TABLA 7: CARACTERÍSTICAS TÉCNICAS DEL TRACKER TOBII

\*Fuente: <http://www.tobii.com/xperience/products/#Specification>



ILUSTRACIÓN 28: TRACKER DE TOBII

### Posibles aplicaciones

- Buscadores Web y lectores pdf pueden desplazarse automáticamente mientras el usuario está leyendo.
- Hacer zoom en mapas mirando un icono.
- Abrir ventanas manteniendo la mirada en esta.
- Jugar a diferentes juegos en primera persona como "shooters" donde el usuario apunta con la mirada y dispara con el ratón.
- Analizar el seguimiento de ojo de usuarios, para el posterior estudio de estos datos y mejorar diseños en páginas web, interfaces, etc, haciéndolos más atractivos.
- Métodos de inicio de sesión como "Windows Hello" mediante reconocimiento facial.
- Arrancar el PC o portátil cuando el sistema detecta al usuario en frente del mismo.
- Encender la pantalla cuando detecta los ojos y apagarse cuando deja de hacerlo.

### EYEGAZE

El dispositivo de seguimiento de ojo EyeGaze pertenece a la empresa LC Technologies, creada en el año 1986 en Virginia, EEUU. Desde sus inicios, ha puesto todos sus recursos en la investigación e implementación de tecnologías de control ocular y eye-tracking. Presenta soluciones completas, software y hardware [21].





ILUSTRACIÓN 29: LOGOTIPO LC TECHNOLOGIES

### Características funcionales

- Necesidad de Windows XP, 7 u 8 como S.O.
- Posibilidad de llevar lentillas y gafas.

### Características técnicas

|                        |                                 |
|------------------------|---------------------------------|
| Frecuencia de muestreo | 60Hz                            |
| Rango de precisión     | 0.45°                           |
| Calibrado              | 5, 9 o 13 puntos                |
| Área de Tracking       | 76cm x 30cm a 60cm de distancia |
| Tamaño de pantalla     | 10" - 35"                       |
| API/SDK                | C++ y C#                        |
| Dimensiones (W/H/D)    | 8.8 x 3.4 x 13.8 (cm)           |
| Conexión               | USB2.0 / USB3.0                 |

TABLA 8: CARACTERÍSTICAS TÉCNICAS DEL TRACKER DE EYEGAZE

\*Fuente: <http://eyegaze.com/wp-content/uploads/EAS%20Monocular%20Technical%20Specifications.pdf>



ILUSTRACIÓN 30: TRACKER DE EYEGAZE EN UN MONITOR

### Posibles aplicaciones

- Jugar a diferentes juegos en primera persona.

- Un teclado en la pantalla donde se puede escribir textos, enviar correos, hablar en chats, etc. Muy útil para personas con discapacidades motoras.
- Analizar el seguimiento de ojo de usuarios, para el posterior estudio de estos datos y mejorar diseños en páginas web, interfaces, etc, haciéndolos más atractivos.
- Generar “Heat maps” y otras estadísticas sobre múltiples materias. Útil para mejorar la usabilidad de páginas web y aplicaciones y para colocación de anuncios.

### Comparativa productos de Eye-tracking

A continuación, se va a llevar a cabo una comparación de los dispositivos descritos anteriormente a través de dos tablas. La primera de ellas se ha realizado basándose en aspectos generales como el sistema operativo soportado, el tipo de licencia, precio, necesidad de hardware adicional, la distancia de funcionamiento del producto y el lenguaje de programación que permite utilizar.

| <u>DISPOSITIVO</u>      | S.O.                      | Licencia                         | Precio (€)    | HW             | Distancia funcionamiento (cm) | Lenguajes             |
|-------------------------|---------------------------|----------------------------------|---------------|----------------|-------------------------------|-----------------------|
| The Eye Tribe [7]       | Android<br>Windows<br>OSX | No disponible                    | 90            | Un dispositivo | 45 - 75                       | C++, C#, Java         |
| SentiGaze [8]           | Windows (PC)              | Ilimitada (un único dispositivo) | No disponible | Ninguno        | 45 - 80 (60 opt.)             | C++, C#, Visual Basic |
| AEye Eye Tracker [9]    | Android<br>Windows        | No disponible                    | 180           | Un dispositivo | 5 a 300                       | C++, C#               |
| Tobii Eye Tracking [10] | Windows                   | No disponible                    | 119           | Un dispositivo | 45 a 100                      | C++, .Net             |
| Eyegaze [11]            | Windows                   | No disponible                    | No disponible | Un dispositivo | 43 - 83                       | C++, C#               |

**TABLA 9: COMPARATIVA GENÉRICA DE DIFERENTES PRODUCTOS DE EYE-TRACKING**

La segunda tabla compara prestaciones técnicas de cada producto como frecuencia de muestreo, rango de precisión, área de tracking, API/SDK y dimensiones.

| <u>DISPOSITIVO</u>      | Frecuencia de muestreo | Rango de precisión | Área de Tracking | Dimensiones |
|-------------------------|------------------------|--------------------|------------------|-------------|
| The Eye Tribe [7]       | <b>VV</b>              | <b>VV</b>          | <b>VV</b>        | <b>X</b>    |
| SentiGaze [8]           | <b>V</b>               | <b>-</b>           | <b>V</b>         | <b>-</b>    |
| AEye Eye Tracker [9]    | <b>VV</b>              | <b>X</b>           | <b>-</b>         | <b>VV</b>   |
| Tobii Eye Tracking [10] | <b>V</b>               | <b>V</b>           | <b>-</b>         | <b>X</b>    |
| Eyegaze [11]            | <b>V</b>               | <b>V</b>           | <b>X</b>         | <b>V</b>    |

**TABLA 10: COMPARATIVA PRESTACIONES TÉCNICAS DE CADA PRODUCTO DE EYE-TRACKING**

Observando las dos tablas comparativas, genérica y técnica, se pueden apreciar las ventajas e inconvenientes de cada uno de los productos. A la hora de elegir el dispositivo más apto para ser utilizado en este trabajo, hay que tener en cuenta una característica que, obligatoriamente, debe cumplir: tiene que soportar Windows, puesto que el caso de estudio que se llevará a cabo se aplicará en dispositivos con este sistema operativo. Por dicho motivo, podremos decidir entre todos los dispositivos estudiados. Al apreciar que el producto “The Eye Tibe” tiene un precio notablemente más bajo y es más pequeño en cuanto a dimensiones, se determina que el caso de estudio se aplicará con ese dispositivo.

### 2.1.2 Técnicas de autenticación basadas en *Eyetracking*

De entre todas las posibles aplicaciones que tiene la tecnología de seguimiento de ojo, se vio que una de ellas es la autenticación en dispositivos. Debido a las mejoras en cuanto a seguridad que es capaz de aportar el eyetracking a un dispositivo y a la necesidad de reforzar las medidas de autenticación, se ha decidido basar este trabajo en el estudio e implementación de una aplicación de inicio de sesión mediante esta tecnología.

Esta aplicación del sistema de seguimiento de ojo aportaría una gran usabilidad al suprimir la necesidad de utilizar un teclado o la mano para realizar la acción. Además, de esta forma se podrían evitar ataques como los de lectura del teclado, observación de cómo se introduce la clave o fijándose en las marcas que pueda dejar la huella de los dedos en un teclado o pantalla.

A continuación, se evaluarán las diferentes técnicas de autenticación que se pueden llevar a cabo con un eye-tracker, para así, analizarlas y poder decidir cuál se utilizará.

#### ***Autenticación basada en movimientos***

Una de las técnicas que se están utilizando para la autenticación en dispositivos o sitios web es la basada en movimientos oculares. Consiste en identificar el lugar exacto al que está mirando el usuario utilizando como forma de inicio de sesión un patrón, un PIN o una contraseña [22].

Este método aportaría una forma fácil y rápida de autenticación, además de ser más segura que las técnicas tradicionales. Por otro lado, seguiría existiendo el problema de la necesidad de recordar una clave, susceptible de ser robada o falsificada.

Esta técnica de autenticación ya ha sido implementada en varias ocasiones y se han realizado pruebas de seguridad y funcionalidad con ella, dando buenos resultados [23].

#### ***Autenticación basada en movimientos y reconocimiento de iris***

Las técnicas de autenticación biométricas basadas en reconocimiento de iris tienen una gran debilidad, ya que, con una simple foto de un iris, se puede ver comprometida su seguridad. Utilizando este método junto al eye-tracking, se puede reducir fuertemente la posibilidad de utilizar este tipo de

ataques al detectar movimiento en el ojo. El dispositivo lee el patrón, PIN o contraseña que el usuario está introduciendo y a la vez realiza un reconocimiento de iris para validar su autenticidad [17].

Esta doble capa de seguridad, a la hora de iniciar sesión, eliminaría el problema de la autenticación basada únicamente en movimientos, ya que, aunque se consiga robar la clave, el iris de cada individuo es diferente. El problema de este método es que cargaría más la máquina computacionalmente, por lo que se necesitaría más potencia que con la primera técnica para conseguir la misma velocidad de acceso.

Se han desarrollado aplicaciones con doble factor de autenticación en los que el primero de ellos (elemento conocido por el usuario) es un PIN, patrón o contraseña y el segundo factor (elemento que, únicamente, posee el usuario) es su iris.

### ***Autenticación basada en la lectura de textos***

Otro método de autenticación es el basado en la lectura de textos, el cual consiste en presentar un texto a un usuario y analizar algunos aspectos, mientras lo lee, como: en qué palabras se para más, cuánto tiempo lo hace, si vuelve a leer algunas palabras, etc. De esta forma, se puede identificar a un individuo con una alta precisión [24].

De esta forma, sería muy difícil imitar la forma de leer un texto de otro sujeto, por lo que en este aspecto presenta una buena seguridad. Su problema reside en que no es una técnica muy usable puesto que el usuario debe leer un texto cada vez que desea iniciar sesión en un dispositivo. Además, cabe tener en cuenta que cuanto más largo es el texto a leer, más seguro es el algoritmo, por lo que el nivel de usabilidad y seguridad serán inversamente proporcionales.

Actualmente, ya existen aplicaciones que permiten autenticar a un usuario basándose en la lectura de textos como, por ejemplo, la creada por Keith Rayner, la cual explica en su artículo “Eye movements in reading and information processing” [24].

### ***Autenticación basada en el movimiento del ojo frente a imágenes***

La última técnica de autenticación que se presenta consiste en presentar al usuario, que quiere iniciar sesión, una o más imágenes y analizar el movimiento de sus ojos mientras las visualiza, a la vez que se mide el tamaño de su pupila en relación a la intensidad de luz de la pantalla [24].

Se han realizado estudios que confirman que cada individuo se fija en puntos diferentes de una figura por lo que este método aportaría seguridad, aunque existiría una posibilidad de coincidencia. Los problemas se encuentran en que no aporta usabilidad al usuario, haciéndose tedioso el inicio de sesión y que, igual que en la técnica anterior (lectura de textos), el nivel de usabilidad y seguridad serán inversamente proporcional, a mayor número de imágenes y variedad, mayor seguridad y menor usabilidad (aumenta el tiempo de acceso).

Se han desarrollado aplicaciones con técnicas parecidas como, por ejemplo, una pantalla llena de diferentes objetos en la cual el usuario debe seguir uno de ellos elegido anteriormente (como contraseña) para poder autenticarse [25]. No obstante, en la revisión de la literatura no se ha encontrado una implementación con estas características.

### Diagrama de las diferentes técnicas de autenticación

A continuación, se representa un diagrama con las distintas técnicas y subtécnicas de autenticación basadas en la tecnología eyetracking.



ILUSTRACIÓN 31: DIAGRAMA DE LAS DISTINTAS TÉCNICAS DE AUTENTICACIÓN

### Conclusión

Las cuatro técnicas de autenticación basadas en la tecnología eyetracking vistas anteriormente, presentan una mayor seguridad y usabilidad frente a las técnicas tradicionales. Puesto que la última de ellas, **autenticación basada en el movimiento del ojo frente a imágenes**, es la única de la que no se han encontrado aplicaciones creadas y es la técnica más novedosa, se ha decidido utilizarla como solución a este trabajo.

Se desarrollará un algoritmo que implemente esta técnica y se buscará un equilibrio entre la seguridad y la facilidad de uso para el usuario, es decir, se utilizará un número suficiente de imágenes, entre las que elegir la contraseña, para que aporte una fuerte seguridad, pero no tan grande como para que haga tediosa y larga la autenticación.

## 2.2 Restricciones y marco regulador

En este apartado, entre otras cosas, se debe analizar el marco legal que afecta al desarrollo del trabajo. En el proyecto desarrollado no aplica hablar de regulación puesto que, no se hace uso en ninguna

ocasión, de datos personales que puedan suponer implicaciones pertenecientes a la Ley Orgánica de Protección de Datos (LOPD).

Por otro lado, se deben analizar las licencias empleadas en la realización de la aplicación. En este caso, solo se utiliza la proporcionada por The Eye Tribe para poder manejar su software. Según exponen en un enlace a GitHub desde su página web, la licencia que incluye el SDK y el servidor *eyetribe* solo consta de limitaciones comerciales. Esto quiere decir, que las aplicaciones y algoritmos desarrollados con este software no pueden ser vendidos. En el caso de que se quisiese comercializar, se podría adquirir una licencia especial que lo permite.

En cuanto a las restricciones, para utilizar la solución propuesta en el TFG es necesario tener el hardware y el servidor de *The Eye Tribe* que cuestan 90 €. Además, para usar el código es requisito tener instalada la versión de *Java 7* o superior. Lo que implica aceptar el acuerdo de licencia de Oracle para dicha versión: “Oracle Binary Code License Agreement for Java SE and JavaFX Technologies”.

Hay que tener en cuenta que exige utilizar un ordenador, no cualquier otro dispositivo. Este ordenador tiene que cumplir unas características:

- PC en versión: Mac, *Mountain Lion 10.8.4*); Windows 8/10.
- Debe tener un puerto USB3.0.

## Capítulo 3: Análisis y diseño de la solución técnica

### 3.1 Especificación de requisitos

En este Trabajo de Fin de Grado se va a estudiar y desarrollar un algoritmo de autenticación basado en la tecnología de seguimiento de ojos. Tras estudiar las diferentes tecnologías de eyetracking y las distintas técnicas de autenticación disponibles, se ha decidido llevar a cabo una aplicación de escritorio que implemente un inicio de sesión basado en el movimiento del ojo frente a imágenes. Para ello, se ha utilizado el dispositivo de eyetracking y el SDK que proporciona la empresa The Eye Tribe.

Con esto se va a crear una herramienta que aporte una mayor seguridad y usabilidad al usuario, a la hora de autenticarse, que las que ofrecían las técnicas tradicionales. Para conseguirlo y evitar que el proceso de inicio de sesión se extienda en exceso y resulte algo tedioso, se va a buscar un equilibrio entre estos dos factores, la seguridad y la usabilidad.

Para llevar a cabo esta aplicación, se deberá crear una interfaz que resulte cómoda y sencilla para el usuario. A la hora de desarrollar esta interfaz será necesario introducir varias ventanas que le permitan realizar la operación básica de inicio de sesión. Además, con el objetivo de conseguir una mayor precisión en la operación de autenticación, se tendrá que introducir un algoritmo de calibrado. Este algoritmo calibrará el tracker en función de la distancia existente entre el usuario y el dispositivo, la distancia entre sus ojos y su posición.

Esa función no es necesario realizarla durante cada autenticación. Cuando el usuario inicie la aplicación el tracker detectará sus ojos con el calibrado por defecto o, en caso de que se hubiese realizado un calibrado anteriormente, utilizará la configuración obtenida del mismo. A pesar de esto, será aconsejable realizar el calibrado siempre que la persona que vaya a autenticarse cambie de postura, posición o el tracker se haya desplazado de su sitio.

También se va a establecer una ventana principal que aparezca al arrancar la plataforma. Esta poseerá una animación que muestra la posición de los ojos del usuario vistos desde el tracker. Con esto, el usuario sabrá si se encuentra cerca o lejos del dispositivo, si tiene la cabeza recta o está demasiado angulada o si necesita centrarse más.

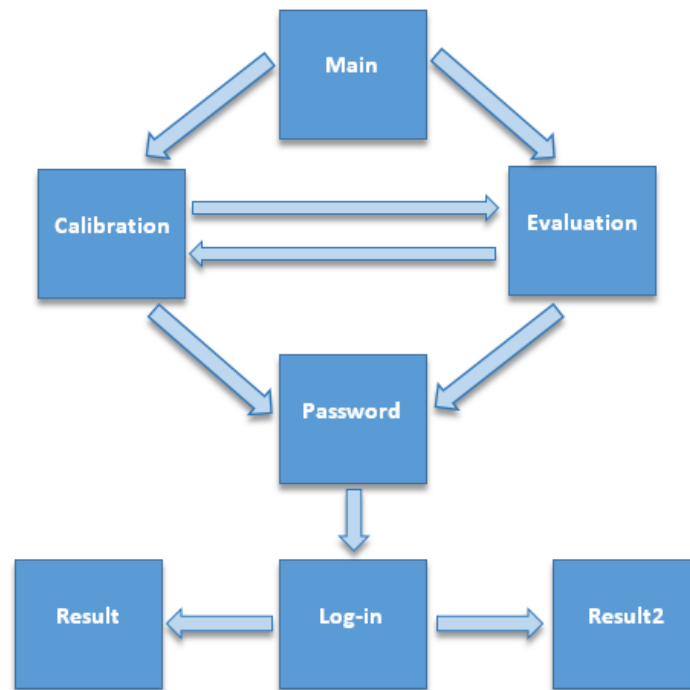
Además, se va a hacer una ventana de evaluación donde el usuario podrá ver si el calibrado se ha llevado a cabo de forma correcta o no.

Otra vista que se incluirá en la aplicación es la de inicio de sesión. En esta, se introducirá el algoritmo de autenticación basado en el movimiento del ojo frente a unas imágenes escogidas de antemano por el usuario como clave. Se buscará que el número de fotos que aparezcan en la ventana sea alto para que el algoritmo aporte una fuerte seguridad, pero no tan elevado como para que deje de ser fácil de usar.

## 3.2 Arquitectura

En este punto se va a explicar la arquitectura que presenta la aplicación de autenticación que propone como solución este trabajo. Se nombrarán y analizarán todas las ventanas posibles dentro de la interfaz de usuario, incluyendo la conexión entre las mismas.

Como se puede ver en el diagrama de la arquitectura que se muestra a continuación (ilustración 19), la aplicación está formada por siete ventanas conectadas entre sí.



**ILUSTRACIÓN 32: ARQUITECTURA DE LA APLICACIÓN**

Las vistas se han diseñado de tal forma que se pudiesen conseguir todos los requisitos planteados en el punto anterior y, además, se han incluido mejoras respecto a estos objetivos al ver, durante el desarrollo de la aplicación, que la harían más funcional.

Se ha diseñado una plantilla que comparten todas las vistas. Esta plantilla está constituida por un fondo gris con los bordes negros, el logotipo de la empresa The Eye Tribe en la parte superior izquierda de la interfaz y un botón con forma de aspa en la parte superior izquierda cuya funcionalidad es la de cerrar la aplicación cuando se pulsa.

A continuación, se va a explicar la estructura y la funcionalidad de cada ventana que forma la aplicación:



- ❖ Al arrancar la aplicación, la primera ventana que se ve es la **principal** (Main). Esta vista tiene una animación en el centro de la pantalla que muestra la posición de los ojos del usuario vistos desde el tracker. También tiene dos botones a través de los cuales se puede acceder a otras dos ventanas. El primero de ellos, “Recalibration”, se encuentra en la esquina inferior izquierda y redirige a la ventana de calibración. El segundo, “Evaluate”, está en la esquina inferior derecha y al pulsarlo envía a la ventana de evaluación.

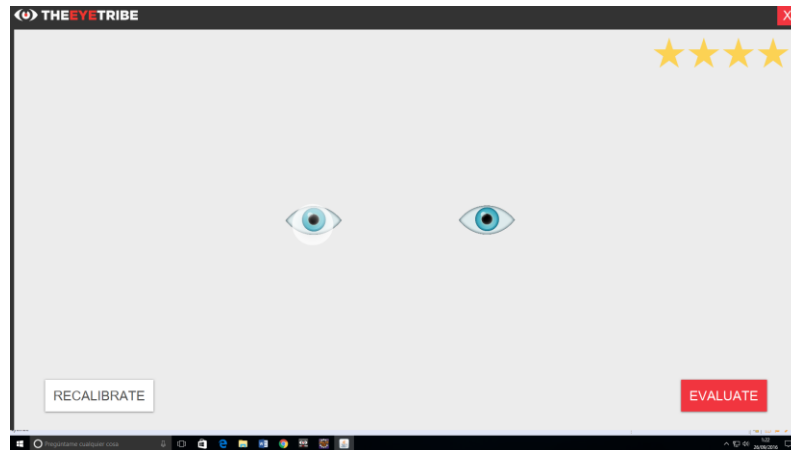


ILUSTRACIÓN 33: VENTANA PRINCIPAL

- ❖ A la página de **calibración** (Calibration) se puede acceder de dos formas, o bien a través de la ventana principal, o bien desde la ventana de evaluación. Una vez se entra en la página, durante unos segundos se muestra un texto indicando que se debe seguir un círculo (“Follow the circle”). Después, de forma intermitente aparece un círculo en distintos puntos de la pantalla, el cual hay que seguir con los ojos para conseguir calibrar el tracker correctamente. Tras un tiempo siguiendo el círculo, el usuario es redirigido automáticamente a la página principal con el dispositivo ya calibrado. Esta función, no obligatoria, se realiza para que la autenticación sea lo más precisa posible, es decir, para que el tracker ajuste la posición del usuario con posición de la pantalla.



ILUSTRACIÓN 34: VENTANA CALIBRATION

- ❖ La ventana de **evaluación** (Evaluation) es accesible únicamente desde la página principal. En esta vista se encuentran dos botones. Uno de ellos, situado en la parte inferior izquierda, redirige a la página de calibrado y el otro ("PASSWORD"), envía al usuario a la ventana donde puede acceder al inicio de sesión. En la zona central de la pantalla se muestran nueve puntos blancos en forma de matriz tres por tres. Cuando el usuario mira a uno de ellos, este se pone de color rojo. La intensidad de este color dependerá de la distancia que exista entre el punto que está mirando el usuario y el centro de los puntos más cercanos. Además, hay un círculo translúcido pintado constantemente en la pantalla que sigue la mirada, lo que resulta muy útil para ver el posible desvío que ha dado la calibración.

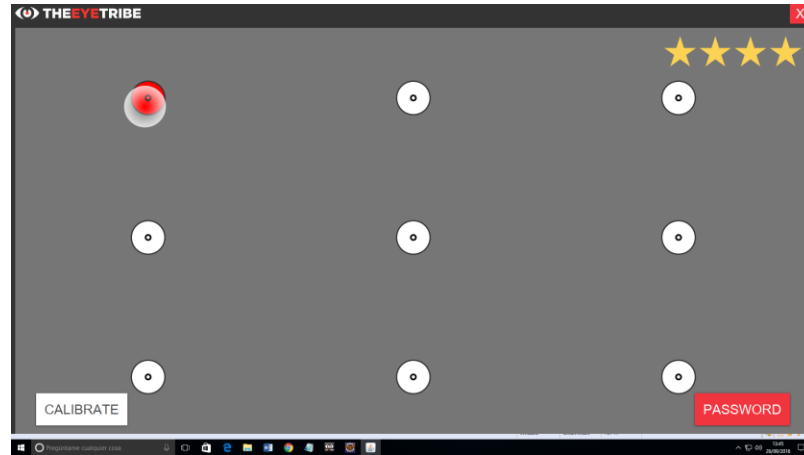


ILUSTRACIÓN 35: VENTANA EVALUATION

- ❖ A la vista en la que se realiza el **inicio de sesión** (log-in) solo se puede llegar a través del botón "Done" que se encuentra en la ventana de evaluación. En la página de autenticación hay un botón de recalibrado, en la parte inferior izquierda, con el objetivo de poder acceder fácilmente a esta función en caso de detectar que el tracker no responde correctamente. En esta página se puede llevar a cabo la función de autenticación. En el centro de la misma, se hayan una serie de fotografías colocadas en forma de matriz. Lo que el usuario necesita hacer para autenticarse es enfocar su mirada en aquellas fotografías que aparezcan en la pantalla y que anteriormente, hayan sido elegidas como clave. Una vez hecho esto, deberá pulsar el botón "Log in", situado en la zona inferior izquierda para comparar el código introducido con el elegido anteriormente. El número de imágenes que se muestran en la pantalla y el número de ellas que se eligen como contraseña, no ha sido seleccionado de forma arbitraria, se ha tenido en cuenta que el objetivo del trabajo es crear una aplicación que aporte tanto seguridad como usabilidad. En el caso de que se quisiese utilizar esta aplicación para proteger información importante y no fuese muy necesario un inicio de sesión rápido, se podría aumentar la cantidad de fotografías mostradas y la cantidad de fotografías que se eligen como contraseña. En el caso contrario, es decir, que se necesitase mayor usabilidad, simplemente habría que reducir el número de ellas.

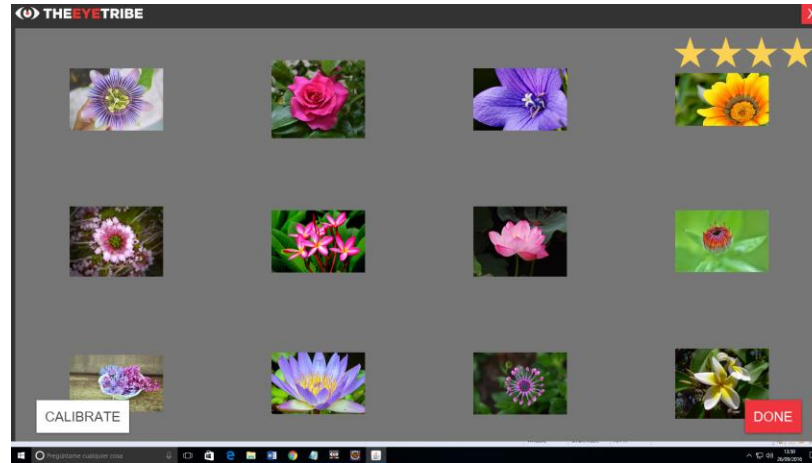


ILUSTRACIÓN 36: VENTANA DE LOG-IN

**Elección de contraseña.** Antes de iniciar la sesión, hay una pantalla en la que se eligen las fotos que se quieren guardar como contraseña. Se muestran un número elevado de imágenes y el usuario debe enfocar su mirada en aquellas que vaya a escoger, en el orden que desee. Una vez realizada la operación, tendrá que pulsar el botón llamado “LOGIN” que le enviará a la página de inicio de sesión. Esta función se ha introducido en la aplicación para poder realizar pruebas con mayor facilidad. En un caso real el usuario solo podría acceder a esta página si fuese la primera vez que configurase la contraseña o si pretendiese cambiarla.

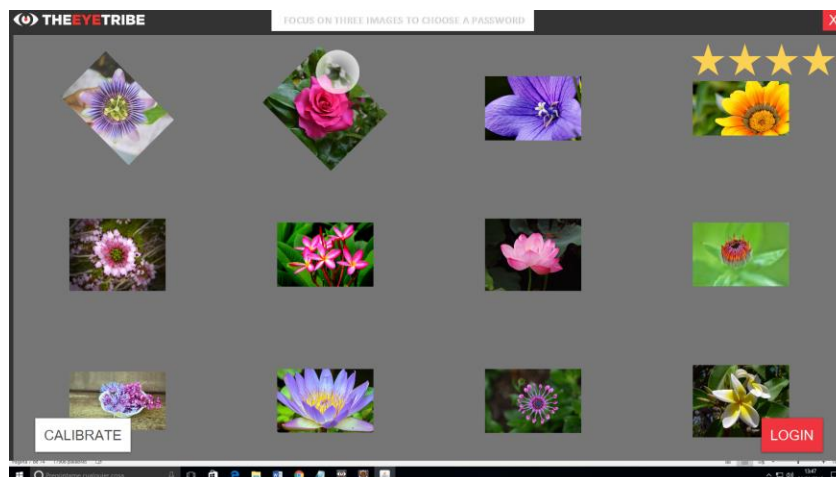


ILUSTRACIÓN 37: VENTANA PASSWORD

- ❖ Las dos últimas ventanas son las que muestran el **resultado de la autenticación** (login result). Si este ha sido satisfactorio, se mostrará un mensaje en el que se informe al usuario de que la introducción de la clave ha sido correcta y la aplicación se dispone a iniciar sesión, “logging in”. En el caso contrario, se muestra el texto “incorrect password”, contraseña incorrecta. Esta página no tiene la opción de redirigir a cualquiera de las otras y la única forma de salir de ella es pulsando

el botón representado como un aspa que cierra la aplicación. En un caso real, si la operación hubiese sido positiva, el usuario accedería automáticamente a la plataforma que esté siendo protegida con esta barrera de seguridad.

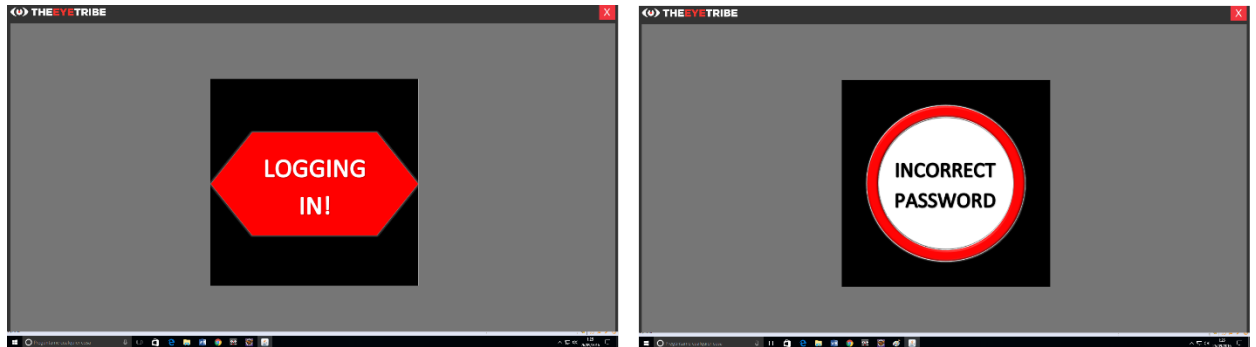


ILUSTRACIÓN 38: VENTANAS DE AUTENTICACIÓN CORRECTA E INCORRECTA

## Capítulo 4: Implementación

En este capítulo se va a explicar cómo se ha realizado la implementación del algoritmo que da como solución, en este Trabajo de Fin de Grado, una aplicación basada en la tecnología de seguimiento de ojos “eyetracking” con la que un usuario puede autenticarse. Se ha buscado crear una herramienta que aporte un alto grado de seguridad y a su vez, sea fácil de usar.

Se va a describir el código desarrollado, desde los paquetes que han sido necesarios crear hasta las clases de las que está formado cada uno de ellos. Se hablará sobre sus funcionalidades, los objetos de los que constan, características principales y, si fuese necesario, también se llevará a cabo una breve explicación de los métodos importantes.

Para el desarrollo del proyecto se ha utilizado principalmente el entorno de desarrollo NeatBeans. En ocasiones, también se ha empleado la herramienta Eclipse, concretamente en su versión “Eclipse Jee Neon”. Aquí se ha creado un proyecto llamado “eyetribes” en el que se han definido los paquetes y clases necesarias, además de importado las librerías que se iban requiriendo a lo largo de la codificación. Para ello, se ha utilizado el lenguaje orientado a objetos *Java* y como librería principal para la representación de la interfaz, *JavaFX*.

La mayoría del código implementado y sus clases han sido definidas por el autor de este trabajo. Algunas de ellas, sobre todo las que se encargan de configurar el servidor, comunicarse con él y recibir los datos como respuesta, se han integrado gracias a un proyecto (*Java*) que la empresa *The Eye Tribe* pone a disposición de todos sus clientes y desarrolladores en la página web *GitHub*. Esto lo hacen con el fin de que todos ellos se puedan centrar en crear aplicaciones y utilidades para su dispositivo lector de ojos y no tengan que codificar la operación de calibrado.

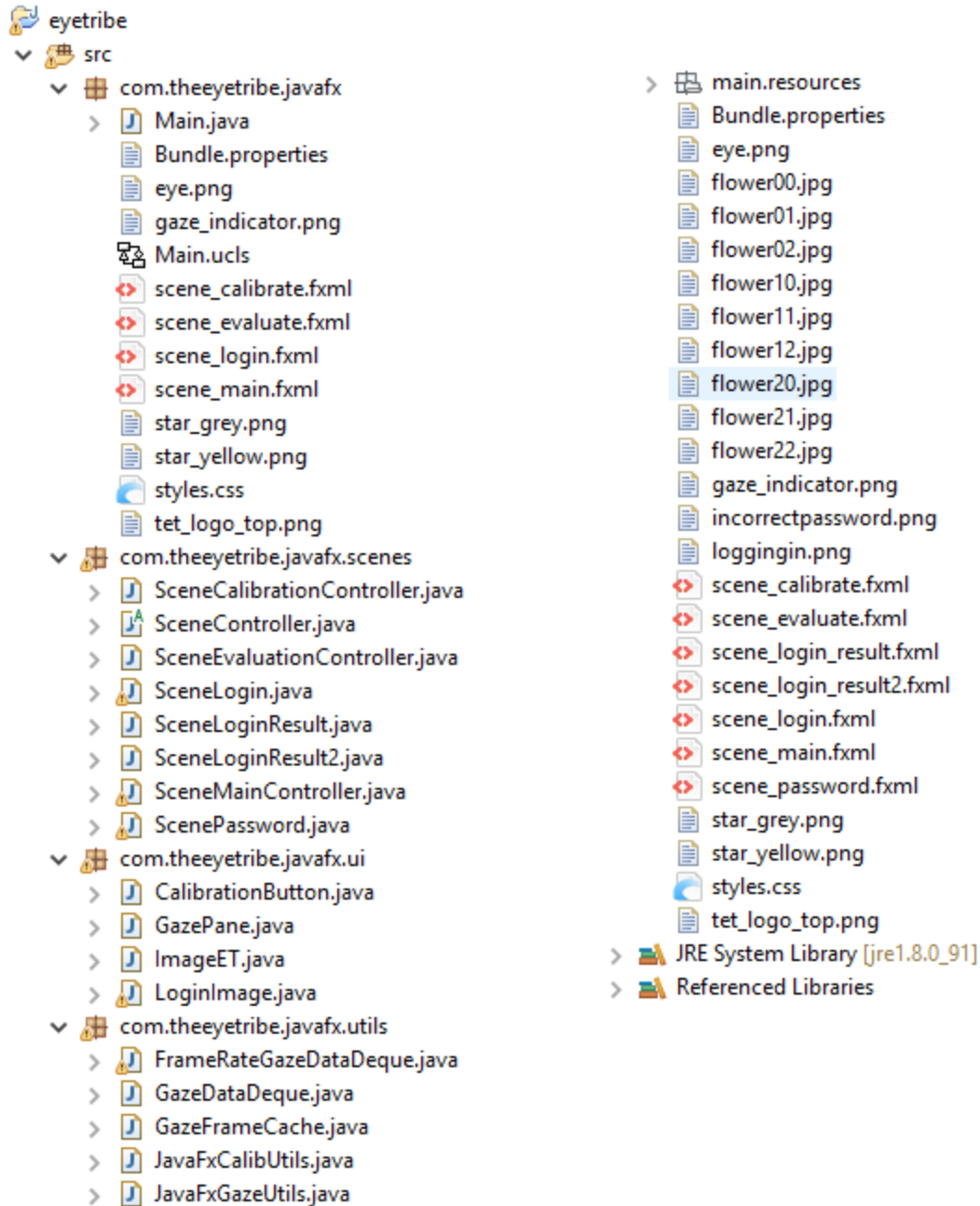


ILUSTRACIÓN 39: ESTRUCTURA DEL PROYECTO EN ECLIPSE

A continuación, se va a proceder a explicar la funcionalidad de cada uno de los paquetes y las clases por las que están formadas:

- ❖ **com.theeyetribe.javaafx:** en este paquete se encuentra la clase “*Main.java*” donde está el método *main* que se encarga de ejecutar todo el código.

- ❖ **com.theyetribе.javafx.scenes:** aquí se encuentran todas las *scenes*. Estas son los controladores que manejan e instancian parámetros para poder enviárselos a los ficheros *fxml* que se encargan de la representación de las ventanas.
- ❖ **com.theyetribе.javafx.ui:** en este paquete se han creado clases que han ido requiriendo cada una de las *scenes*.
- ❖ **com.theyetribе.javafx.utils:** todas las clases transversales para todo el código desarrollado se han instanciado en este paquete, clases que se encargan de detectar y analizar la mirada del usuario, las que se comunican con el servidor enviando peticiones y recogiendo las respuestas, etc.

Una vez definidos los paquetes que forman el proyecto, se va a proceder a nombrar y explicar cada una de las clases que los constituyen.

#### Paquete “com.theyetribе.javafx”:

Como se ha comentado anteriormente, solo está formado por la clase ejecutable “*Main.java*”.

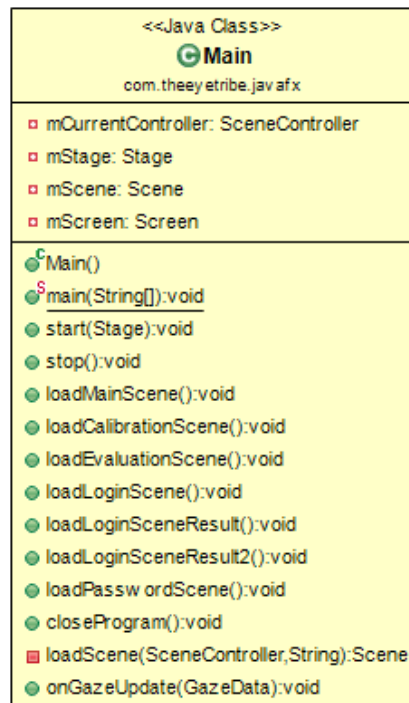


ILUSTRACIÓN 40: ESTRUCTURA DE LA CLASE **MAIN**

#### ❖ **Main.java**

En esta clase se encuentra el método “*main*” el cual se encarga de ejecutar todo el proyecto. Además de este, también hay métodos importantes como:

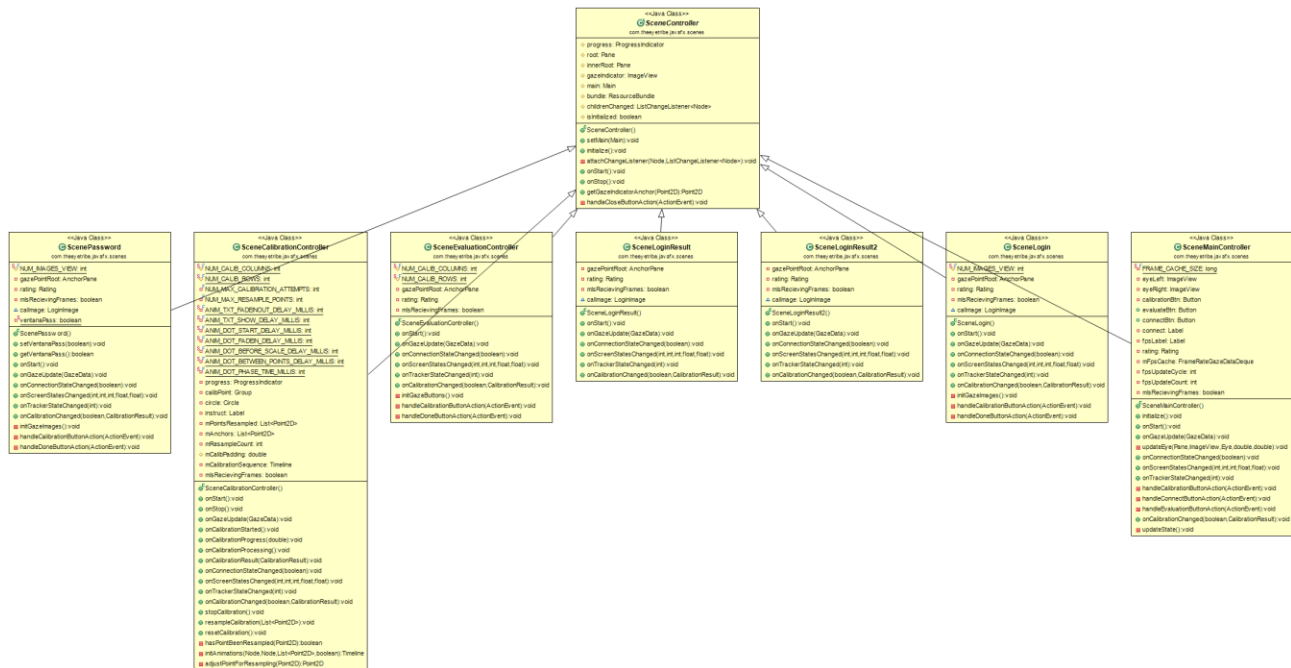
- **start:** este método tiene como misión arrancar todo el programa, es decir, iniciar la interfaz principal, sincronizar el código con el servidor y pasarle parámetros como la versión del API. Asimismo, llama al método encargado de

seguir el movimiento del ojo y establecer las coordenadas de sus ojos con respecto a las de la pantalla.

- ***stop***: detiene la impresión de las ventanas o vistas y después, corta toda comunicación con el servidor.
- ***closeProgram***: cierra el programa de forma brusca.
- ***loadScene***: es el controlador que se encarga de levantar la escena que le invoca. Cuando se le llama, este cierra la ventana que estuviese abierta, crea una nueva e instancia todos los controladores y configuraciones necesarias.
- ***onGazeUpdate***: actualiza los datos relacionados con el seguimiento del ojo mientras el programa está corriendo.

### Paquete “com.theyetribе.javafx.scenes”:

Este paquete lo forman el controlador de las escenas y todas las escenas. El objetivo de estas es el de manejar las vistas de la aplicación y definir los elementos que les darán forma como botones, animaciones y paneles.



### ILUSTRACIÓN 41: ESTRUCTURA DEL PAQUETE *SCENES*

❖ *SceneController.java*

Esta es la clase base de todos los controladores de las escenas, es decir, aquella que contiene los métodos comunes. Los más importantes son:



- **initialize:** configura e inicializa los *listeners* que se necesitan en todas las ventanas disponibles y los sincroniza con ellas.
- **handleCloseButtonAction:** es el manejador del evento que se acciona al pulsar el botón de cierre de aplicación (representado en la interfaz con forma de aspa). En él se invoca al método *closeProgram* que cierra la aplicación.

#### ❖ **SceneMainController.java**

*SceneMainController* tiene como función dirigir y modelar la **vista principal**. Sus métodos más importantes son:

- **onGazeUpdate:** actualiza los datos relacionados con el seguimiento del ojo mientras el programa está corriendo. Llama a la función *updateEye* y pinta la animación de los ojos en la pantalla en función de la posición del usuario, si tiene los dos abiertos o solo uno, el ángulo de inclinación de su cabeza, etcétera. También define las configuraciones necesarias para la visualización del puntero transparente, que sigue la trayectoria de la mirada del individuo, en la pantalla.
- **updateEye:** recoge los datos y las coordenadas de los ojos para después, añadirlos a una variable *ImageView*.
- **handleCalibrationButtonAction:** es el manejador del evento que se acciona al pulsar el botón “Calibration” y envía al usuario a la ventana de calibrado.
- **handleConnectButtonAction** es el manejador del evento que se acciona al pulsar el botón “Connect”. Este botón aparece cuando el servidor no ha sido arrancado. Una vez hecho esto se deberá pulsar este botón para acceder a la página principal
- **handleEvaluationButtonAction:** redirige a la ventana de evaluación al ser pulsado.

#### ❖ **SceneCalibrationController.java**

Esta clase es el controlador de escenas (*SceneController*) asociado a la calibración del tracker. En ella se configuran los parámetros necesarios para la representación de la calibración y los botones de la página. Igualmente, se recogen los datos obtenidos y se interpretan para posteriormente enviarlos al servidor.

Los métodos más importantes de los que consta la clase son:

- **onCalibrationResult:** maneja los valores obtenidos de la calibración y toma decisiones. Si la calibración ha sido exitosa el programa redirige al usuario a la página de evaluación e introduce los resultados en la variable que se encarga de almacenar los datos de la calibración. En caso de que se haya producido algún error, se detiene la calibración y se carga la página principal de la aplicación.

- **initAnimations:** este método define los parámetros relacionados con el calibrado y los necesarios para su visualización.

#### ❖ **SceneEvaluationController.java**

Esta clase sirve para manejar y configurar la pantalla que se encarga de la evaluación del calibrado. Sus principales métodos son los siguientes:

- **onGazeUpdate:** actualiza los datos relacionados con el seguimiento del ojo mientras el programa está funcionando. En ella se invoca a la clase *initGazeButtons* y se define las configuraciones necesarias para la visualización del puntero transparente, que sigue la trayectoria de la mirada del individuo, en la pantalla.
- **initGazeButtons:** define, estructura y modela los círculos que se representan en la ventana de evaluación, número de ellos, anchura, altura, posicionamiento, etc.
- Además, consta de los manejadores de botones. Uno para el que redirecciona la página al calibrado (**handleCalibrationButtonAction**) y otro para el que redirecciona a la página donde se elige la contraseña (**handleDoneButtonAction**).

#### ❖ **ScenePassword.java**

Esta clase es el controlador asociado a la pantalla en la que se elige la contraseña. Para hacerlo se debe enfocar la mirada en las tres imágenes que se quieran guardar como clave en el orden que se desee. Sus métodos más importantes son:

- **initGazeImages:** este método es muy parecido al *initGazeButtons* de la clase *SceneEvaluationController* salvo que, en vez de configurar círculos, lo hace con las imágenes que se mostrarán como posible clave.
- **onGazeUpdate:** realiza la misma función que el resto de métodos con este mismo nombre, pero invocando a *initGazeImages*.
- También hay dos métodos manejadores de eventos llamados como los de la clase *SceneEvaluationController*, **handleCalibrationButtonAction** y **handleDoneButtonAction**, que redirigen a la ventana de calibración y a la de inicio de sesión, respectivamente.

#### ❖ **SceneLogin.java**

La clase *SceneLogin* se encarga de manejar la vista de inicio de sesión. Su estructura y visualización es muy parecida a la de la clase *ScenePassword*, diferenciándose únicamente en la redirección de sus botones y en los datos que lee y almacena de las acciones que va realizando el usuario con la mirada.

- **handleDoneButtonAction:** este método en función de cuál haya sido el resultado de la autenticación, el cual comprueba con el método *getSuccess* de la clase *LoginImage*, redirige al usuario a una clase u otra. Si ha sido exitosa,

se le envía a la clase *SceneLoginResult* y, en caso contrario, a la clase *SceneLoginResult2*.

❖ ***SceneLoginResult.java***

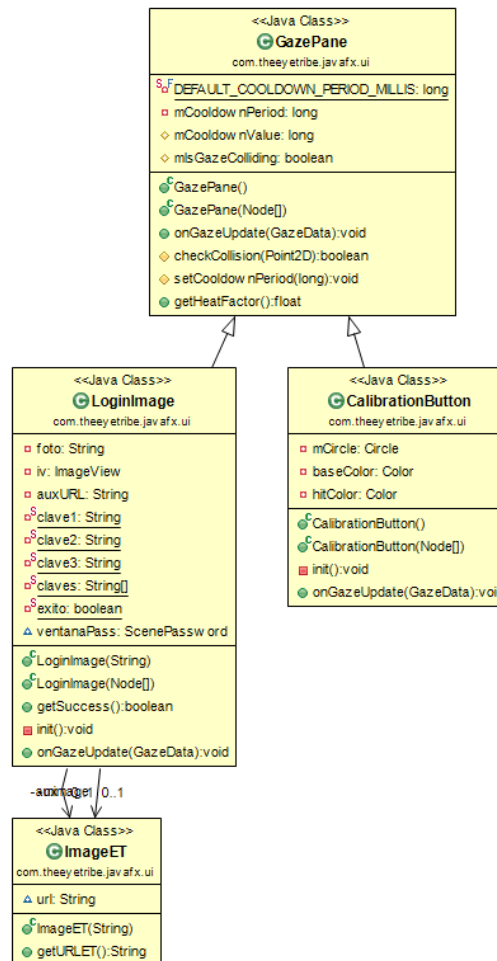
Esta clase simplemente sirve para soportar la ventana en la que se le indica al usuario que la contraseña que ha introducido es correcta y que se está iniciando sesión.

❖ ***SceneLoginResult2.java***

Esta última clase únicamente realiza de soporte a la vista en la que se le informa al usuario de que la clave introducida es incorrecta.

**Paquete “com.theyetribes.javaafx.ui”:**

Este paquete está formado por aquellas clases que han ido necesitando los paquetes anteriores y algunas clases de este. Estas clases son:



**ILUSTRACIÓN 42: ESTRUCTURA DEL PAQUETE UI**

#### ❖ **CalibrationButton.java**

Esta clase inicializa y declara un componente con forma de círculo que cambia de color con el movimiento de los ojos. Esta clase la utilizan los ficheros “.java” *SceneCalibrationController* y *SceneEvaluationController* para representar en sus vistas los círculos que requieren.

Los métodos más importantes que lo forman son:

- **init:** este método crea dos objetos de la clase *Circle* y los personaliza dándoles color, tamaño, forma, etc. Después, con la clase *Group* los añade a un grupo para que otras clases puedan trabajar con él. Además, junta los dos círculos para que se visualicen uno dentro del otro.
- **onGazeUpdate:** actualiza los datos relacionados con el seguimiento del ojo mientras el programa está corriendo. En este caso, llama a una función que aporta una tonalidad de rojo a los círculos pintados, que depende de la distancia que hay entre el punto que el usuario está mirando y el centro de ese círculo. El color será más intenso cuanto menor sea esa distancia.

#### ❖ **GazePane.java**

Esta clase reconoce el movimiento de los ojos del usuario y, a partir de unos métodos que crea le da un color determinado a otro objeto. Esta clase la utiliza *CalibrationButton*.

Sus métodos principales son:

- **checkCollision:** este método comprueba si hay colisión entre el punto que está mirando el usuario y los círculos mostrados en la pantalla y devuelve *true* si la hay y *false*, si no.
- **getHeatFactor:** devuelve el “valor de calor”, que es un indicador de cuánto es la distancia entre el punto al que está mirando el usuario (representado en la pantalla por un círculo translúcido) y los círculos representados en la pantalla.

#### ❖ **ImageET.java**

*ImageET* extiende de la clase *Image* (*javafx.scene.image.Image*). Fue creada puesto que en un método de la clase *LoginImage* se necesitaba conseguir la URL que tenía un objeto de la clase *Image*, pero la función “*getURL*” de esta estaba obsoleta y no devolvía su valor.

Su único método es:

- **getURLET:** devuelve el valor de la URL que tiene el objeto *ImageET*.

#### ❖ **LoginImage.java**

Esta clase define un componente con forma de imagen que realiza una serie de acciones en función de la pantalla en la que se encuentre el usuario y de lo que esté enfocando con su mirada. La utilizan las clases *SceneLogin* y *ScenePassword*.

Los métodos más importantes son:

- **init:** crea un objeto de tipo *ImageET*, lo almacena en uno de tipo *ImageView* y personaliza este dándole tamaño a la foto. Después, con la clase *Group* añade

el objeto *ImageView* a un grupo para que otras clases puedan trabajar con él y representarlo.

- **getSuccess:** este método simplemente devuelve un *boolean* que indicará si la autenticación ha tenido éxito o no, es decir, si el usuario ha introducido la contraseña correcta.
- **onGazeUpdate:** actualiza los datos que se relacionan con el seguimiento del ojo mientras el programa se está ejecutando. En el caso de que sea la clase *ScenePassword* quien haya inicializado un objeto *LoginImage*, esta función leerá las imágenes en las que el usuario está enfocando su mirada y las almacenará como clave de acceso. Si por el contrario es la clase *SceneLogin* quien ha creado el objeto *LoginImage*, las imágenes que vaya enfocando con la mirada serán comparadas con las elegidas como contraseña.

### Paquete “com.theyetribе.javafx.utilс”:

Este paquete está constituido por clases transversales, es decir, que utilizan y necesitan todas las clases de los demás paquetes

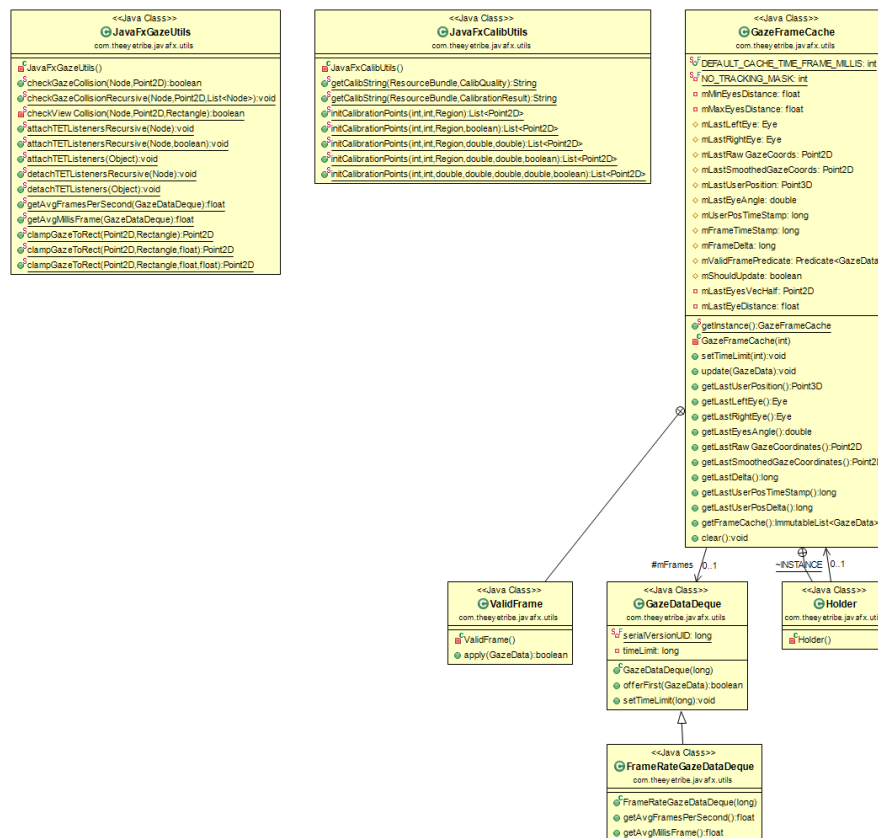


ILUSTRACIÓN 43: ESTRUCTURA DEL PAQUETE UTILS

❖ ***GazeDataDeque.java***

Esta clase almacena el último valor del objeto *GazeData*, el cual tiene la información que le envía el tracker al servidor.

❖ ***FrameRateGazeDataDeque.java***

Extiende de la clase *GazeDataDeque* y calcula el valor de la tasa de fotogramas (“frame rate”).

❖ ***GazeFrameCache.java***

El objetivo de esta clase es evitar los pequeños fallos técnicos que pueda ocasionar un mal seguimiento del ojo por parte del usuario, provocando un envío de datos erróneos. Para ello, mantiene almacenados, durante un tiempo, datos correctos relacionados con los fotogramas.

- ❖ Las clases ***JavaFxCalibUtils.java*** y ***JavaFxGazeUtils.java*** definen métodos que se encargan de dar soporte a las funciones de calibrado y seguimiento del ojo.

Una vez explicado el contenido de cada uno de los paquetes, se va a proceder a realizar una breve descripción de los ficheros con extensión “.fxml”. Estos se encargan de la visualización de cada una de las ventanas disponibles en la aplicación, cogiendo algunos datos de las clases “Scene” creadas.

Desde estos ficheros se crean y personalizan algunos de los parámetros de la interfaz de usuario como, por ejemplo, el color gris del fondo, el marco negro de la pantalla, el estilo y fuente de los botones, las imágenes introducidas como el logotipo de *The Eye Tribe*.

A continuación, se muestra una tabla en la que se representa la asociación que hay entre cada fichero de visualización y su controlador.

| <b><u>FICHERO DE VISUALIZACIÓN (.fxml)</u></b> | <b><u>SCENE CONTROLLER (.java)</u></b> |
|--|--|
| <i>scene_main</i>                              | <i>sceneMainController</i>             |
| <i>scene_calibrate</i>                         | <i>SceneCalibrationController</i>      |
| <i>scene_evaluate</i>                          | <i>SceneEvaluationController</i>       |
| <i>scene_password</i>                          | <i>ScenePassword</i>                   |
| <i>scene_login</i>                             | <i>SceneLogin</i>                      |
| <i>scene_login_result</i>                      | <i>SceneLoginResult</i>                |
| <i>scene_login_result2</i>                     | <i>SceneLoginResult2</i>               |

**TABLA 11: ASOCIACIÓN DEL FICHERO DE VISUALIZACIÓN CON EL CONTROLADOR**

## Capítulo 5: Evaluación y resultados

### 5.1 Dificultades en el desarrollo

En este punto se van a describir los principales problemas que se han ido planteando durante el desarrollo del proyecto, en especial, en la etapa de la implementación.

La mayoría de estas dificultades han sido solventadas con éxito, pero en algunos casos, se han tenido que buscar otras alternativas al plan inicial previsto para poder seguir avanzando.

La empresa *The Eye Tribe* dispone en su página web de desarrolladores de un enlace que redirecciona a GitHub donde se encuentra una librería *Java*. Esta librería, creada con el objetivo de que los programadores se puedan centrar en inventar aplicaciones que den funcionalidad al tracker, proporciona una interfaz *Java* de comunicación con el servidor y un algoritmo de calibración.

El primer problema encontrado y el más costoso se produjo al descargar el archivo *JAR* y las librerías indicadas en la página web. Cuando se procedió a importar el proyecto para arrancar la herramienta, se observó que al ejecutarlo aparecían varios errores en el código. Se estuvo intentando detectar esos errores un largo periodo de tiempo. Después, se vio en un foro de desarrolladores que proporciona *The Eye Tribe* comentarios diciendo que la última versión disponible tenía errores que estaban intentando solucionar y en estos, se encontraban enlaces al proyecto en otras versiones anteriores. Finalmente, se probó una versión anterior importando sus librerías correspondientes y la aplicación se arrancó correctamente.

Otra de las dificultades que se fueron encontrando, fue el hecho de familiarizarse con el código que proporcionaba el proyecto de GitHub y con el API de las librerías “*eyetribe-java-0.6.60.jar*” y “*EyeTribeJavaFx.jar*”. El problema de la primera de ellas se debe a que consta de muchos paquetes con clases y métodos complejos. En cuanto a la segunda, nunca se había trabajado con la tecnología *Java Fx* y se hacía difícil entender las funciones. Todos estos problemas fueron solventados dedicando el tiempo necesario a su estudio.

Por último, al compilar el código creado y trabajar con la aplicación, en algunas ocasiones aparece un error “*NullPointerException*” en la consola del entorno de desarrollo. Esto no se ha conseguido solucionar, pero tras llevar a cabo muchas pruebas con la plataforma se ha visto que no afecta de ninguna manera a su correcto funcionamiento.

### 5.2 Pruebas realizadas

Una vez se ha tenido la aplicación de autenticación acabada y funcionando correctamente, se han llevado a cabo dos tipos de pruebas. A continuación, se van a nombrar las pruebas que se han llevado a

cabo, después se realizará una breve explicación contando en qué han consistido y finalmente, se evaluarán y describirán los resultados obtenidos:

### **Utilización de gafas de vista**

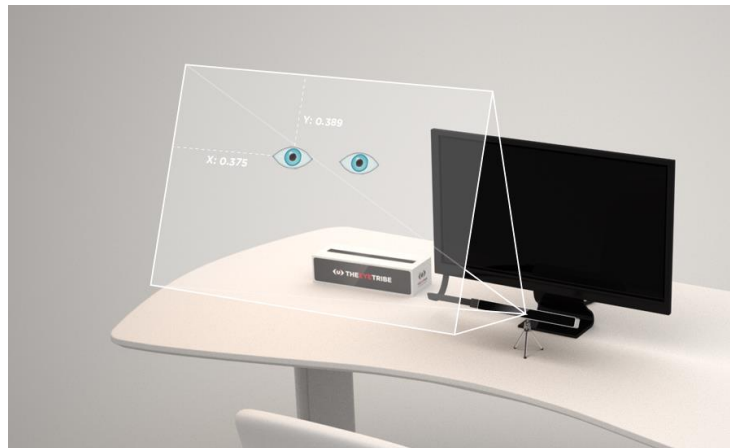
Se decidió comprobar si la aplicación también podría ser utilizada por gente que necesitase gafas graduadas para poder ver. Para ello, se puso a un usuario con gafas delante del dispositivo lector de ojos y se arrancó la plataforma.

La primera vez que se realizó la prueba, se observó en la vista principal de la aplicación que el tracker no detectaba bien los dos ojos. En ocasiones la animación que enseña la posición de la mirada del usuario en la pantalla, solo mostraba uno de los ojos o ninguno. Tras posicionar la cabeza del individuo en varios ángulos se supo que el problema era el marco de las gafas, impidiendo este que las luces provenientes del dispositivo se reflejasen en el ojo.

Sabido esto, se volvió a llevar a cabo la prueba pero situando la cabeza del usuario en un ángulo en el que el marco de las gafas no interfiriese y el resultado fue satisfactorio, la herramienta funcionó correctamente.

### **Distancia del usuario con respecto al tracker**

Otra de las pruebas que se ha realizado consiste en verificar la distancia de funcionamiento máxima y mínima del tracker. Para calcular el área y la distancia de funcionamiento según la posición del usuario, el dispositivo describe un plano rectangular como se puede ver en la ilustración siguiente:



**ILUSTRACIÓN 44: ÁREA Y DISTANCIA DE FUNCIONAMIENTO DEL TRACKER**

Para llevar a cabo esta prueba, se ha situado a un individuo delante del dispositivo con la página principal de la aplicación abierta y se ha observado la animación de los ojos.

Primero se ha probado con la distancia máxima posible para que esta animación deje de funcionar, aproximadamente 90 cm. Se ha podido realizar la autenticación, pero con muchas dificultades.



Después, se ha hecho lo mismo utilizando la distancia que las especificaciones técnicas del tracker describen como máxima, 75 cm. El inicio de sesión se ha llevado a cabo de forma correcta, pero se observa que no es la distancia óptima.

Una vez conocida la distancia máxima de funcionamiento se han hecho las pruebas para detectar la mínima. Se ha examinado con la distancia mínima necesaria para que la animación de la vista principal deje de funcionar, 45 cm aproximadamente. En esta ocasión, la autenticación se ha podido hacer sin ningún problema.



## Capítulo 6: Gestión del Proyecto

### 6.1. Planificación

Para ilustrar la planificación del Proyecto presentado en este TFG se va a utilizar una herramienta de software libre llamada Gantt Project con la que se ha realizado un Diagrama de Gantt. Dicho diagrama es un esquema en el que se muestran todas las tareas que componen el trabajo y el tiempo de dedicación previsto para cada una de ellas, identificando sus fechas de inicio y fin.

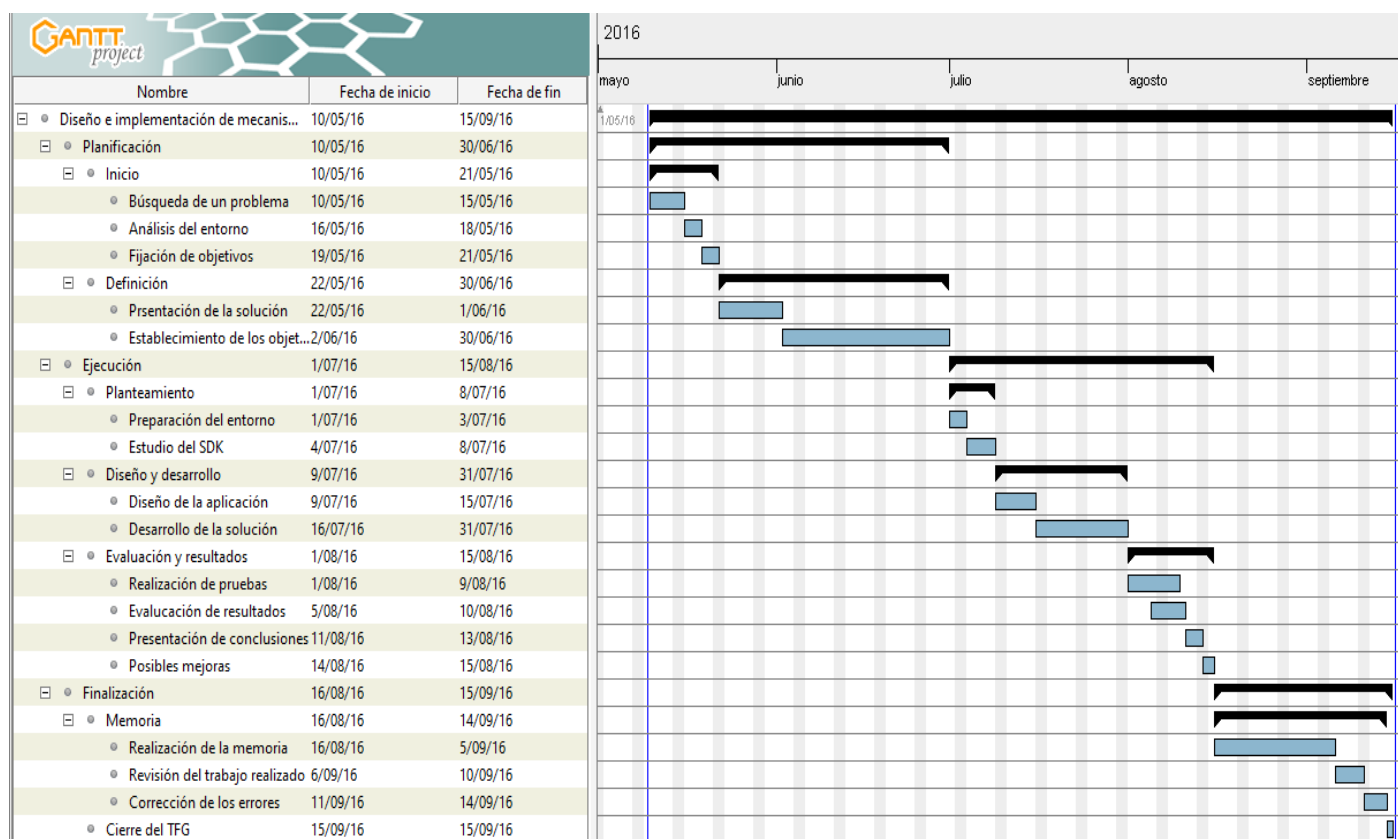


ILUSTRACIÓN 45: DIAGRAMA GANTT DEL TFG

Este trabajo comenzó el día 10 de mayo de 2016 con la etapa de **Planificación**, la cual está formada por las siguientes partes:

**Planificación:** Desde el día 10 de mayo de 2016 hasta el 30 de junio de 2016.

- **Inicio** → Comienza el día 10 de mayo y acaba el 21 de mayo.
  - Búsqueda de un problema.
  - Análisis del entorno a estudiar.

- Fijación de objetivos.
- **Definición** → Comienza el día 22 de mayo y acaba el 30 de junio.
  - Presentación de una solución al tutor.
  - Establecimiento de los objetivos.

La segunda etapa del Proyecto es la **Ejecución**. Es la más larga puesto que comprende desde el planteamiento del trabajo hasta la evaluación de los resultados, pasando por el diseño y desarrollo del mismo. Está compuesta por las siguientes partes:

**Ejecución:** Desde el día 1 de julio de 2016 hasta el 15 de agosto de 2016.

- **Planteamiento** → Comienza el día 1 de julio y acaba el 8 de julio.
  - Preparación del entorno de trabajo.
  - Estudio del SDK de Java.
- **Diseño y desarrollo** → Comienza el día 9 de julio y acaba el 31 de julio.
  - Diseño de la aplicación final.
  - Desarrollo de la solución.
- **Evaluación y resultados** → Comienza el día 1 de agosto y acaba el 15 de agosto.
  - Realización de diferentes pruebas.
  - Evaluación de los resultados obtenidos de las pruebas.
  - Presentación de conclusiones.
  - Planteamiento de posibles mejoras en el sistema.

La última etapa es la **Finalización** que consiste en la realización y entrega de la Memoria. Las partes por las que está formada son:

**Finalización:** Desde el día 16 de agosto de 2016 hasta el 15 de septiembre de 2016.

- **Memoria** → Comienza el día 16 de agosto y acaba el 14 de septiembre.
  - Realización de la Memoria.
  - Revisión del trabajo por parte de la tutora.
  - Corrección de los errores.
- **Cierre del Trabajo de Fin de Grado** → Se produce el día 15 de septiembre de 2016.

## 6.2 Presupuesto

En este apartado se va a realizar un desglose de los costes que produce llevar a cabo este proyecto. Se va a basar en las diferentes tareas que se han producido a lo largo del mismo y se procederá a describir, de cada una de dichas etapas, el intervalo de tiempo en el que han transcurrido (en días) y la cantidad de horas que se han dedicado (en horas).

| TAREA  | INTERVALO DE TIEMPO |
|--------|---------------------|
| Inicio | 12 días             |

|                                |         |
|--------------------------------|---------|
| <b>Definición</b>              | 40 días |
| <b>Planteamiento</b>           | 8 días  |
| <b>Diseño y desarrollo</b>     | 23 días |
| <b>Evaluación y resultados</b> | 15 días |
| <b>Finalización</b>            | 31 días |

**TABLA 12: INTERVALO DE TIEMPO TRANSCURRIDO POR TAREA**

Para calcular el número de horas de trabajo que se han invertido en cada una de las tareas se tendrán en cuanto tanto los días lectivos como los festivos. Si se desean más detalles, leer el apartado anterior “6.1. Planificación”.

| <b>TAREA</b>                   | <b>HORAS DEDICADAS</b>      |
|--------------------------------|-----------------------------|
| <b>Inicio</b>                  | 10 días x 5h/día            |
| <b>Definición</b>              | 40 días x 2h/día            |
| <b>Planteamiento</b>           | 8 días x 5h/día             |
| <b>Diseño y desarrollo</b>     | 23 días x 5h/día            |
| <b>Evaluación y resultados</b> | 15 días x 5h/día            |
| <b>Finalización</b>            | 31 días x 5h/día            |
| <b>TOTAL</b>                   | 129 días = <b>460 horas</b> |

**TABLA 13: CANTIDAD DE HORAS DEDICADAS POR TAREA**

En total se han destinado 500 horas de trabajo a la realización de este Proyecto.

## ■ **COSTES DEL MATERIAL**

Para llevar a cabo el cálculo de los **costes materiales**, se tiene en cuenta tanto un ordenador con capacidad y potencia suficientes para soportar una máquina virtual, las licencias correspondientes al software y el dispositivo de seguimiento de ojos de la empresa “*The Eye Tribe*” como todos los recursos necesarios para la ejecución del Proyecto tales como electricidad para el funcionamiento de los equipos, como conexión a Internet y páginas relacionadas para realizar la tarea de estudio e investigación. Por todo ello, se tienen los conceptos descritos en la siguiente tabla:

| <b>CONCEPTO</b>                                    | <b>Coste</b> | <b>Dedicación (meses)</b> | <b>Vida útil (meses)</b> | <b>COSTE imputable</b> |
|--|--------------|---------------------------|--------------------------|------------------------|
| <b>Ordenador x64 – Intel Core i7 – CPU 3.60GHz</b> | 900 €        | 4                         | 40                       | 90 €                   |

|  |       |   |    |              |
|--|-------|---|----|--------------|
| Licencia Windows 10 Pro                  | 270 € | 4 | 36 | 30 €         |
| Licencia Microsoft Office 2015           | 100 € | 3 | 12 | 25 €         |
| Dispositivo de seguimiento de ojos + SDK | 90 €  | 4 | 24 | 15 €         |
| Gastos de electricidad                   | 70 €  | - | -  | 70 €         |
| Conexión a Internet – 300Mbps            | 120 € | - | -  | 120 €        |
| <b>COSTE TOTAL DEL MATERIAL</b>          |       |   |    | <b>350 €</b> |

TABLA 14: DESGLOSE DE LOS COSTES MATERIALES DEL PROYECTO

### ▪ COSTES DEL PERSONAL

A la hora de calcular los **costes del personal**, se va a considerar que los participantes en el Proyecto son dos, el autor del Trabajo de Fin de Grado y la doctora Tutora del mismo.

Con el fin de describir estos datos, a continuación se muestra una tabla basada en la categoría, el tiempo invertido en la realización (en horas) y el sueldo (en Euros) de la persona.

| CATEGORÍA                       | COSTE (€/h) | TIEMPO (h) | TOTAL           |
|---------------------------------|-------------|------------|-----------------|
| Dra. Ingeniera                  | 60          | 50         | 3.000 €         |
| Graduado                        | 30          | 460        | 13.800 €        |
| <b>COSTE TOTAL DEL PERSONAL</b> |             |            | <b>16.800 €</b> |

TABLA 15: DESGLOSE DE LOS COSTES PERSONALES DEL PROYECTO

### ▪ COSTE TOTAL DEL PROYECTO

Finalmente, se lleva a cabo una tabla que representa el **coste total** del Proyecto con la suma de los costes materiales, personales e impuestos.

| CONCEPTO                        | PRECIO          |
|---------------------------------|-----------------|
| Costes del material             | 350 €           |
| Costes del personal             | 16.800 €        |
| <b>COSTE TOTAL DEL PROYECTO</b> | <b>17.150 €</b> |

TABLA 16: DESGLOSE DE LOS COSTES TOTALES DEL PROYECTO

El coste total de la realización completa del Proyecto es de **17.150 €**

## Capítulo 7: Conclusiones y líneas futuras

### 7.1 Conclusiones

En este Trabajo de Fin de Grado, se pretendía estudiar e implementar un algoritmo de autenticación basado en la tecnología eyetracking y crear una plataforma para iniciar sesión. Se eligió esta nueva tecnología al ser innovadora y al aportar una gran usabilidad y seguridad a las técnicas de inicio de sesión. Esto se propuso como solución a al ver que las técnicas tradicionales no respondían a las necesidades actuales en cuanto a seguridad y facilidad de uso.

Además, se eligió el método de autenticación basado en el movimiento de los ojos de un individuo (situado delante del dispositivo tracker) frente a unas imágenes. Esto se decidió observando una comparativa de las diversas técnicas más innovadoras en la actualidad. Lo que esta comparación mostró es que, aparte de proporcionar mayor seguridad y usabilidad que el resto, era la más reciente y, según la revisión de la literatura llevada a cabo, no se ha encontrado una implementación con estas características.

Para realizar diseño la aplicación, primero se estuvo analizando los requerimientos del sistema, viendo cuántas ventanas se necesitaban y cómo tenían que ser. Después se estudió las clases y métodos que iban a requerir cada una de ellas, con el objetivo de buscar la arquitectura más óptima.

Teniendo en cuenta los conocimientos de programación del autor de este proyecto y que el *SDK Java API* era el más sencillo de los que ofrecía *The Eye Tribe*, se decidió utilizar el lenguaje de programación orientado a objetos *Java*.

Una vez desarrollado el trabajo utilizando el diseño codificado y realizadas las pruebas correspondientes, se ha observado que la aplicación funcionaba correctamente cumpliendo todas las especificaciones requeridas previamente.

Asimismo, una vez se ha terminado el trabajo y se ha conocido más a fondo la tecnología, se ha podido percibir que existen posibles modificaciones que mejorarían la solución propuesta.

### 7.2 Líneas futuras

Como se ha visto en el punto anterior, la aplicación desarrollada cumple con todos los objetivos iniciales previstos. Sin embargo, una vez adquiridos los conocimientos suficientes acerca de la tecnología, el dispositivo, el entorno de desarrollo y el lenguaje de programación, se observa que existen nuevas funcionalidades o mejoras que podrían llevarse a cabo en un futuro. A continuación, se presentan algunas de ellas:

La primera mejora y la más importante, es aumentar la **seguridad** de la técnica de autenticación desarrollada mediante el uso de un mayor número de imágenes en la página de autenticación. De esta forma, las probabilidades de acertar las imágenes elegidas como contraseña por el usuario se reducirían. Otra opción para mejorar la seguridad, es utilizar el método de autenticación desarrollado en este TFG más de una vez y después, comparar si las imágenes escogidas estaban guardadas como contraseñas. Estas dos opciones son compatibles entre sí y juntas, disminuirían notablemente las probabilidades de intrusión.

Otra futura mejora puede ser implementar la aplicación para otros **sistemas operativos** compatibles con el SDK como Mac OS X. Además, se pueden realizar versiones para dispositivos móviles como smartphones o tablets siempre que tengan instalado el sistema operativo Android.

Con el objetivo de confirmar que la técnica de autenticación elegida es mejor que las demás estudiadas, se podría implementar un algoritmo basado en alguna de ellas. Así, después se podría **comparar** los resultados en cuanto a usabilidad y seguridad que proporcionan.

Con el propósito de generalizar la aplicación, se ha elegido como idioma el inglés. Una medida futura a tomar es ampliar el número de idiomas implementándola, por ejemplo, en **francés y español**, para hacerla accesible a un mayor grupo de personas.



## Referencias

- [1] U. Jendricke, D. Gerd, & T. Markotten. "*Usability meets security-the Identity-Manager as your personal security assistant for the Internet.*" Computer Security Applications, 2000. ACSAC'00. 16th Annual Conference. IEEE, 2000.
- [2] "Producto: Microsoft Office Word 2016". Disponible online: <https://products.office.com/es-es/word> (Último acceso: agosto 2016).
- [3] M. Merino. "*¿Qué son los mapas de calor?*", TICbeat Blog, 2014. Disponible online: <http://www.ticbeat.com/tecnologias/que-son-los-mapas-de-calor/> (Último acceso: agosto 2016).
- [4] The Eye Tribe. "*Eyetribe Products*". Disponible online: <https://theeyetribe.com/products/> (Último acceso: agosto 2016).
- [5] O. B. Fernández, "*Introducción al lenguaje de programación Java. Una guía básica*", 2005. Disponible online: <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf> (Último acceso: agosto 2016).
- [6] GENBETA:dev. "*NetBeans*", 2014. Disponible online: <http://www.genbetadev.com/herramientas/netbeans-1> (Último acceso: agosto 2016).
- [7] The Eye Tribe. "*Eyetribe Products*". Disponible online: <https://theeyetribe.com/products/> (Último acceso: agosto 2016).
- [8] NeuroTechnology. "*Sentigaze technical specifications*". Disponible online: <http://www.neurotechnology.com/sentigaze-technical-specifications.html> (Último acceso: agosto 2016).
- [9] EyeTech. "*Technical specifications for AEye eye Tracking*". Disponible online: <http://www.eyetechds.com/board-level-oem.html> (Último acceso: agosto 2016).
- [10] Tobii. "*Tobii Xperience Specification*". Disponible online: <http://www.tobii.com/xperience/products/#Specification> (Último acceso: agosto 2016).
- [11] EyeGaze. "*EyeGaze Monocular Technical Specifications*". Disponible online: <http://eyegaze.com/wp-content/uploads/EAS%20Monocular%20Technical%20Specifications.pdf> (Último acceso: agosto 2016).
- [12] Roper-Hall, G. (2007). Historical Vignette Louis Émile Javal (1839–1907): The Father of Orthoptics. *American Orthoptic Journal*, 57(1), 131-136.
- [13] Linguist-Educator Exchange. "*The History of the Science of Reading: Huey and the Psychology of Reading*". Disponible online: <https://linguisteducatorexchange.com/library/history-of-reading-instruction/the-history-of-the-science-of-reading-huey-and-the-psychology-of-reading/> (Último acceso: agosto 2016).
- [14] Pernice, K., & Nielsen, J. (2009). How to conduct eyetracking studies. *Nielsen Norman Group*, Fremont, CA.
- [15] NSU. "*Eye-Tracking en Interacción Persona-Ordenador*". Disponible online: <http://www.nosolousabilidad.com/articulos/eye-tracking.htm> (Último acceso: agosto 2016).
- [16] The Eye Tribe. "*Eyetribe Basics*". Disponible online: <http://dev.theeyetribe.com/general/> (Último acceso: agosto 2016).

- [17] The Eye Tribe. "THE EYE TRIBE STORY". Disponible online: <https://theeyetribe.com/the-eye-tribe-story/> (Último acceso: septiembre 2016).
- [18] NeuroTechnology. "Company Information". Disponible online: <http://www.neurotechnology.com/about.html> (Último acceso: septiembre 2016).
- [19] EyeTech. "Our Company". Disponible online: <http://www.eyetechds.com/about-eyetech.html> (Último acceso: septiembre 2016).
- [20] Tobii. "About Tobii". Disponible online: <http://www.tobii.com/group/about/> (Último acceso: septiembre 2016).
- [21] LC Technologies. "Our Story". Disponible online: <http://www.eyegaze.com/our-story/> (Último acceso: septiembre 2016).
- [22] Smartmatic. "Nuevo método de autenticación biométrica basado en el movimiento ocular". Disponible online: <http://biometria.smartmatic.com/nuevo-metodo-de-autenticacion-biometrica-basado-en-el-movimiento-ocular> (Último acceso: septiembre 2016).
- [23] R. Bednarik, et al. "Eye-movements as a biometric." Scandinavian conference on image analysis. Springer Berlin Heidelberg, 2005.
- [24] D. Maltoni, & A. Jain (Eds.). "Biometric Authentication: ECCV 2004 International Workshop, BioAW 2004", Prague, Czech Republic, May 15, 2004, Proceedings. Vol. 3087. Springer Science & Business Media, 2004.
- [25] Liu, D., Dong, B., Gao, X., & Wang, H. (2015, June). Exploiting Eye Tracking for Smartphone Authentication. In *International Conference on Applied Cryptography and Network Security* (pp. 457-477). Springer International Publishing.

